

Technická univerzita v Liberci

Hospodářská fakulta

Technická univerzita v Liberci

Hospodářská fakulta

INFORMAČNÍ SYSTÉM HOSPODÁŘSKÉ FAKULTY

INFORMATION SYSTEM OF THE FACULTY OF ECONOMICS

DIPLOMOVÁ PRÁCE

2000-2001

Jakub Steiner

doc. Ing. Petr Štěpánek

Ing. Vladimír Záděra

Technická univerzita v Liberci

Hospodářská fakulta

Obor Podniková Ekonomika

I N F O R M A Č N Í S Y S T É M H O S P O D Á Ř S K É F A K U L T Y

I N F O R M A T I O N S Y S T E M O F T H E F A C U L T Y O F E C O N O M I C S

DP-PE-KPE-99044

Jakub Steiner

Vedoucí práce:

doc. Ing. Josef Sixta, Csc. – HF TU v Liberci

Konzultant:

Ing. Vladimíra Zádová – KIN HF TU v Liberci

Počet stran: 62

Počet příloh: 1

23.05.99

TECHNICKÁ UNIVERZITA V LIBERCI

Hospodářská fakulta

Katedra podnikové ekonomiky

Školní rok 1998/99

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro **Jakuba Steinera**

obor č. 6268 - 8 Podniková ekonomika

Vedoucí katedry Vám ve smyslu zákona č. 172 / 1990 Sb. o vysokých školách a navazujících předpisů určuje tuto diplomovou práci:

Název tématu: **Informační systém Hospodářské fakulty**

Zásady pro vypracování:

Ve své diplomové práci se zaměřte na analýzu současného IS Hospodářské fakulty TU v Liberci (výchozí stav k 1.9.1997) s ohledem na jeho aktualizaci.

Zpracujte návrhy řízení aktualizace IS s více pracovišti tak, aby tento návrh byl realizovatelný na jiných fakultách. V závěru posuďte ekonomickou rozvahu Vámi navrženého způsobu aktualizace IS na fiktivním uživateli z výrobní sféry.

1/57/99 H

KPE/POE
61.5.1.5.2. příl.

RESUMÉ

Tato diplomová práce pojednává o problematice vývoje informačního systému v organizaci. Praktická část popisuje všechny fáze životního cyklu projektu nového subsystému IS Hospodářské fakulty TU v Liberci až po implementaci nového systému. Zmíněny jsou některé nejběžnější metody, používané v jednotlivých fázích vývoje systému.

Závěr práce nabízí porovnání dvou metod pořízení nového informačního systému – vlastní vývoj, nebo nákup hotového „balíku“. Zároveň je zmíněna problematičnost použití klasických kapitálových rozhodovacích metod posuzování investic do projektů na pořízení IS.

In my dissertation, I mainly focus on the IS development lifecycle. Every step of SDLC is documented on the actual Faculty of Economics IS sub-system development project that started mid. 1998. Both the tools and techniques are discussed. In every step of the cycle, I only focus on the methods used most often.

In the final part the difficulties of using standard capital appraisal techniques on IS investments are explained. Also, two most common ways of acquiring a new IS are discussed – in-house development project and purchased package.

POUŽITÉ TERMÍNY, ZKRATKY

Datagram	IP paket. Pro přenos dat je v rámci IP vrstvy síťového modelu TCP/IP používán blok dat – datagram.
DBMS	Systém řízení báze dat. Software pro manipulaci s databázemi.
Extranet	Neveřejná podniková síť v prostředí globální sítě Internet. Neveřejnost zabezpečena pomocí kryptovacích technologií.
Firewall	Prostředek pro zabezpečení sítě před externím útokem.
Intranet	Neveřejná podniková síť používající standardy použité v globálním Internetu.
IP	Internet protokol. Protokol z rodiny TCP/IP starající se o adresování a nezaručený přenos datagramů v prostředí internetu.
IS	Informační systém.
PHP	Professional Hypertext Preprocessor. Skriptový jazyk pro vytváření dynamických dokumentů pro službu WWW.
SDLC	System Development LifeCycle – Životní cyklus vývoje IS.
SQL	Standard Query Language. Standard pro formulování dotazů na databázi.
TCP	Transport control protocol – Standard z rodiny protokolů TCP/IP zabezpečující „bezpečný“ přenos dat.
UI	User Interface - Uživatelské rozhraní.

OBSAH

1. ÚVOD	11
1.1 SYSTÉM	11
2. FÁZE TVORBY IS	12
2.1 ŽIVOTNÍ CYKLUS VÝVOJE IS (SDLC)	12
2.2 DEFINOVÁNÍ PROBLÉMU	12
STUDIE PROVEDITELNOSTI	13
2.4 ANALÝZA	14
2.4.1 Metody získání dat	15
2.4.2 Funkční analýza	17
2.4.3 Datové modelování	18
2.4.4 Entita	18
2.4.5 Atribut	19
2.4.6 Vztahy	19
2.4.7 Datový slovník	19
2.5 NÁVRH (DESIGN)	19
2.6 REALIZAČNÍ FÁZE	20
2.7 PŘECHOD NA NOVÝ SYSTÉM	20
2.8 ÚDRŽBA	20
2.9 SYSTÉMY CASE	22
2.9.1 Upper/Front-End CASE	22
2.9.2 Middle CASE	22
2.9.3 Lower CASE	22
3. INTRANET	23
3.1 MODULARITA	25
3.2 EXTRANET	25
4. TECHNICKÉ PROSTŘEDKY	27
4.1 HTML	27
4.2 PHP	27
4.3 SQL	28
4.4 DATABÁZOVÝ SYSTÉM MYSQL	28

4.5 BEZPEČNOST	30
4.5.1 Uživatel.....	31
4.5.2 WWW server.....	31
5. INFORMAČNÍ SYSTÉM HOSPODÁŘSKÉ FAKULTY	36
5.1 DEFINOVÁNÍ PROBLÉMU	36
5.2 STUDIUM PROVEDITELNOSTI	37
5.3 ANALÝZA	39
5.3.1 Konceptuální model	39
5.3.2 Logická struktura datové základny.....	40
5.3.3 Katedra	42
5.3.4 Studijní program	43
5.3.5 Předmět	43
6. POPIS SYSTÉMU	45
6.1 AUTORIZACE.....	45
6.2 PŘIDAT ZAMĚSTNANCE	46
6.3 PŘIDAT PŘEDMĚT.....	47
6.4 ODEBRAT ZAMĚSTNANCE.....	47
6.5 ODEBRAT PŘEDMĚT	47
6.6 OPRAVIT ZAMĚSTNANCE.....	48
6.7 OPRAVIT PŘEDMĚT	48
7. PODNIKOVÝ IS	49
7.1 IS JAKO KONKURENČNÍ VÝHODA	49
7.2 KONTINUITA HODNOTY IS/IT	49
7.2.1 Základní hodnota	50
7.2.2 Inovační hodnota.....	50
7.3 STRATEGIE POUŽITÍ IS/IT	50
7.4 NÁKLADY IS	51
7.5 HODNOCENÍ INVESTIC DO IS	54
7.5.1 Techniky pro kapitálové posouzení investic	54
7.6 KOUPIT NEBO VYROBIT	57
7.7 NÁKUP HOTOVÉHO „BALÍKU“	57
7.8 VLASTNÍ VÝVOJ.....	58
8. ZHODNOCENÍ.....	60

POUŽITÁ LITERATURA61

Černý, J. (2005). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Dvořák, M. (2006). Úvod do informačních systémů. Brno: Vydavatelství VŠE.

Horáček, J. (2007). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Jakub, S. (2008). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Štěpán, P. (2009). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Černý, J. (2010). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Dvořák, M. (2011). Úvod do informačních systémů. Brno: Vydavatelství VŠE.

Horáček, J. (2012). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Jakub, S. (2013). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Štěpán, P. (2014). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Černý, J. (2015). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Dvořák, M. (2016). Úvod do informačních systémů. Brno: Vydavatelství VŠE.

Horáček, J. (2017). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Jakub, S. (2018). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Štěpán, P. (2019). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Černý, J. (2020). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Dvořák, M. (2021). Úvod do informačních systémů. Brno: Vydavatelství VŠE.

Horáček, J. (2022). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Jakub, S. (2023). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Štěpán, P. (2024). Informační systémy v podniku. Praha: Vydavatelství VŠE.

Černý, J. (2025). Informační systémy v podniku. Praha: Vydavatelství VŠE.

1. Úvod

Hospodářská fakulta je relativně samostatná část Technické univerzity v Liberci. Smyslem mé diplomové práce nebylo vytvoření plošného IS Technické univerzity v Liberci, ba dokonce ani Hospodářské fakulty. Zaměřil jsem se čistě na subsystém komunikace fakulty se studenty jednotlivých programů na ní vyučovaných.

I tak, na první pohled úzká oblast, vyžadovala daleko více času, než jsem zpočátku předpokládal a za normálních okolností by bylo zapotřebí minimálně dvou osob v realizačním týmu. Je pouze vyjímečné a dokonce nežádoucí, aby analytik byl zároveň programátorem.

1.1 Systém

Pro účely této práce si musíme vymezit některé základní pojmy. Systémem budeme chápat skupinu vzájemně souvisejících komponent které slouží společnému účelu. V dalším textu se setkáme s pojmem informační systém a systém organizace. Systémem organizace¹ chápeme specifický systém jehož smyslem je dosáhnout určitých podnikatelských cílů. Každé vedení organizace by ve své podnikatelské strategii, plánu, mělo rozvést své cíle, aby mohlo posoudit nakolik systém organizace splňuje svou úlohu tyto cíle plnit.

Informační systém pracuje s daty, která potřebuje organizace. O IS můžeme hovořit jako o podsystému podnikového systému (business system). IS existuje pouze proto, aby sloužil systému organizace, jehož je součástí. Existuje mnoho definic vymezení pojmu IS. Informační systém se obvykle skládá ze čtyř základních prvků – báze dat, hardware, software a komunikace.

V tradičním světě bylo informačních systémů využíváno pouze jako nástroje výkaznictví. IS byl využit jako nástroj sledování toho, co se v organizaci v minulosti stalo. Tedy něco co je nutné, ale s ohledem na cíle organizace funguje IS pouze jako nutnost, nikoli jako nástroj zlepšení systému organizace. IS ale může být použit jako nástroj zlepšení efektivity v organizaci. Některé ilustrační příklady použití IS v organizaci uvádím v kapitole 7.1 - IS jako konkurenční výhoda.

¹ Organizace ve smyslu podnik - business system. V případě Hospodářské fakulty bude snazší hovořit o organizaci než o podniku.

2. FÁZE TVORBY IS

Systémová analýza a návrh (design) jsou termíny používané v souvislosti s tvorbou nebo úpravou IS, který již organizaci nestačí. Organizace již existující systém „přerostla“. Pakliže tento stav nastane, organizace se snaží právě pomocí těchto postupů problém řešit.

Analýzou nazýváme fázi, kdy jsou zjišťovány požadavky na nově vznikající IS. Fáze návrhu spočívá ve vytvoření schémat, plánů nového systému.

2.1 Životní cyklus vývoje IS (SDLC)²

Každý IS prochází několika fázemi svého vývojového „životního cyklu“ (system development life cycle). Tento ustálený termín zahrnuje následujících 7 položek:

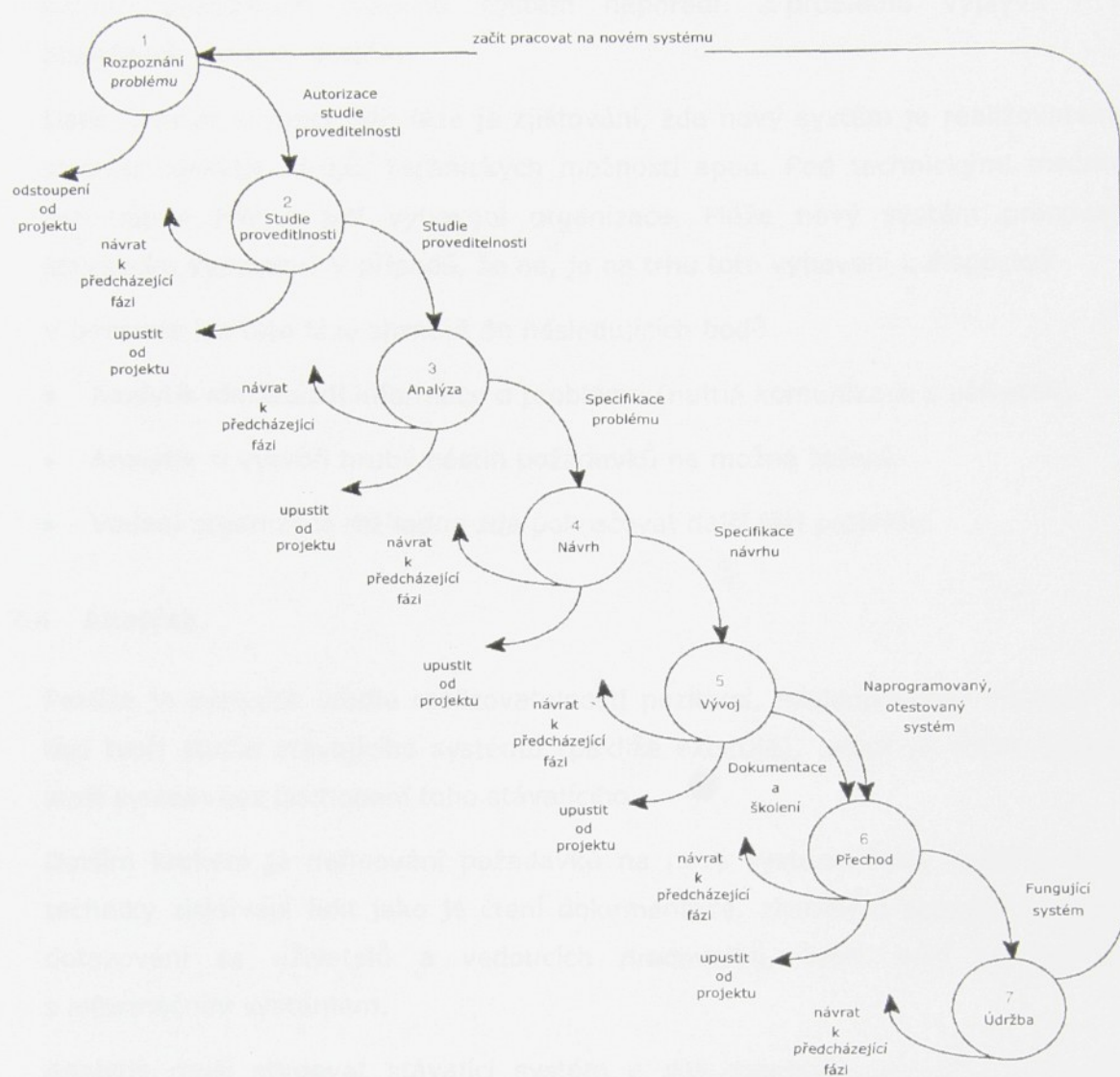
1. Definování problému.
2. Studie proveditelnosti (feasibility study).
3. Analýza.
4. Návrh (design).
5. Vlastní tvorba.
6. Přechod.
7. Údržba.

Vztah mezi jednotlivými fázemi popisuje schéma obrázku 1 na následující straně.

2.2 Definování problému

Zrod nového systému zaznamenáme, když vedoucí pracovníci organizace přestanou být spokojeni se stávajícím systémem a nalézají nedostatky v IS. Starý IS již nereflektuje hlavní funkce organizace. Například mohlo dojít k významnému rozšíření výroby, nebo v našem případě ke zvýšení počtu uchazečů o studium. Důvodem mohou být i nedostatečné možnosti rozšíření funkcí systému pro budoucí rozvoj.

² Kendall, P.A.: Introduction to Systems Analysis: A Structured Approach, WCB 1989, str. 7-23



obrázek 1 - životní cyklus IS

V případě že rozdíl mezi požadavky a tím, co stávající IS poskytuje je markantní, může vedení organizace pověřit systémového analytika, aby se problémem podrobněji zabýval.

2.3 Studie proveditelnosti

Systémový analytik v této fázi přichází s tzv. feasibility study. Cílem této fáze je definovat přesně problém z bodu 2.2 a s minimálním úsilím a prostředky stanovit zda je možné vytvořit nový systém. Tato fáze většinou netrvá déle než jeden měsíc.

Systémový analytik posuzuje, co je přesně důvodem vzniku problému, proč si s tímto problémem stávající systém neporadí. Z problému vyplývá i nový požadavek na nový systém.

Další činností v rámci této fáze je zjišťování, zda nový systém je realizovatelný po stránce lidských zdrojů, technických možností apod. Pod technickými možnostmi rozumějme HW a SW vybavení organizace. Může nový systém pracovat na stávajícím vybavení? V případě, že ne, je na trhu toto vybavení k dispozici?

V podstatě lze tuto fázi shrnout do následujících bodů:

- Analytik shromáždí informace o problému (nutná komunikace s uživateli).
- Analytik si vytvoří hrubý nástin požadavků na možná řešení.
- Vedení organizace rozhodne zda pokračovat další fází projektu.

2.4 Analýza

Pakliže je výsledek studie realizovatelnosti pozitivní, následuje fáze analýzy. Tuto fázi tvoří studie stávajícího systému (pakliže existuje), neboť je těžké navrhnout lepší systém bez pochopení toho stávajícího.

Dalším krokem je definování požadavků na nový systém. Tady používá analytik techniky získávání fakt jako je čtení dokumentace, zkoumání stávajících procesů, dotazování se uživatelů a vedoucích pracovníků, kteří mají co do činění s informačním systémem.

Analytik musí studovat stávající systém a dokumentovat ho, aby byl schopen porozumět jednak jeho slabým ale i silným stránkám. Analytik by si měl vytvořit seznam požadavků na nový IS. Měl by si vytvořit schémata funkcí nového systému bez toho, aby rozváděl, jak tyto funkce zabezpečit. Cílem této fáze může být vytvoření prototypu systému. Smyslem tohoto prototypu je umožnit uživatelům „osahat“ si systém ještě před jeho dokončením a nalézt problémy, které se většinou zjistí až teprve potom, co s ním uživatelé dostatečně dlouhou dobu pracují.

Klíčovou metodou v této fázi jsou interview s vedoucími pracovníky a uživateli systému, neboť oni nejlépe znají problémy a svoje požadavky na nový IS.

Na konci této fáze analytik upraví původní předpoklady ze studie proveditelnosti. Tak, jak se vyvíjí analytikova znalost původního systému, měly by všechny odhady reflektovat tyto změny.

2.4.1 Metody získání dat

Základní metodou je již zmíněné studium existující dokumentace. Tato dokumentace se může skládat ze specifikace postupů, popisu uživatelského prostředí, souborů, programů. Jinými slovy se jedná o zdroje dat popisující to, jak byl systém navržen, aby fungoval.

Jinou stránkou věci je, jak skutečně systém funguje. Došlo během vývoje organizace ke změnám v dokumentaci? Metodami získání potřebných znalostí jsou například pozorování zaměstnanců při práci se systémem, nebo snaha systém sám používat. Mezi hlavní metody ale patří dotazování se pomocí osobního interview, nebo zprostředkovaně pomocí dotazníku.

2.4.1.1 Dotazník

Dotazník je metodou získávání dat od poměrně velkého počtu dotazovaných. Jedná se o jednosměrný způsob komunikace, kdy předmětem otázek jsou současné procedury a možná zlepšení.

Důvodem použití dotazníků je zpravidla množství dotazovaných a jejich geografické roztroušení, takže by pro analytika nebylo možné se dotazovat osobně.

Dotazníky ale mohou být použity společně s osobním interview. V této kombinaci je možné použít interview jako doplněk k dotazníku, kdy se analytik ptá na sporné oblasti a upřesňuje si tak nejasnosti.

Formulovat správně dotazník, který by vedl k navrácení spolehlivých, korektních dat je velice složité. Z důvodů zpracování vrácených dotazníků jsou otázky většinou formulované „černobíle“, zatímco respondenti většinou chtějí vyplnit „něco mezi“.

Před vlastní formulací otázek si analytik musí vyjasnit, kdo bude dotazník vyplňovat a jakou informaci chce od dotazovaných získat. Prvním krokem je tedy vytvoření seznamu otázek, na které chce pomocí dotazníku získat odpovědi. Každá z otázek musí cíleně mířit na konkrétní informaci, kterou chce analytik získat.

Každý dotazník by měl být zaslán společně s doprovodným dopisem. V něm by měl být identifikován autor dotazníku, kdo analýzu inicioval a koho kontaktovat v případě nesrovnalostí. Zároveň by dopis měl obsahovat stručný popis, jak s dotazníkem pracovat, dokdy a kam má respondent vyplněný dotazník zaslat.

Vzhled dotazníku do určité míry předurčuje, jaké odpovědi analytik dostane zpět. Dobře napsaný a přehledný dotazník umožní respondentům vyplnit pečlivě

promyšlené odpovědi. V dotazníku by měl být použit čitelný, dostatečně veliký řez písma. Ideálně by měly být otázky a popis otázek odlišen rozdílným řezem a velikostí písma. Otázky by neměly být nahuštěny na sebe z důvodů šetření místem na stránce. Celkově by mělo být použito dostatečně prázdného místa, aby respondent nabyl pocitu, že se jedná o jednoduchý dotazník. Žádná otázka by neměla přesahovat z jedné na další stránku. Co se týče rozsahu, záleží na předmětu dotazníku a zainteresovanosti respondenta na daném tématu. Když respondent odpovídá na otázky které se ho určitým způsobem týkají, většinou mu ani pětistránkový dotazník nedělá problémy. Pakliže ovšem dotazník pojednává o věcech všeobecných, i dvoustránkový dotazník může připadat dlouhý.

Otázky týkající se stejného tématu by měly být seskupeny. Otázky, které se jeví jako důležité z pohledu respondenta je vhodné umisťovat na začátek dokumentu. To zabezpečí, že respondent bude dotazníkem hned na počátku zaujat a zvyšuje to pravděpodobnost návratu dotazníku.

Jedním z nejdůležitějších rozhodnutí, v souvislosti s dotazníkem, je posouzení, zda použít anonymní nebo podepsané dotazníky. V případě anonymních dotazníků se často vrátí pravdivější odpovědi, když se dotazník zabývá kontroverznějším tématem v organizaci. Na druhou stranu podepsané dotazníky umožňují kontaktovat respondenta v případě, kdy je potřeba objasnit nejasnou odpověď.

Z důvodů hromadného zpracování je v dotaznících používáno jednak uzavřených otázek nebo škál. Někdy je možné využít omezené otevřené otázky (třeba pro zjištění konkrétní hodnoty – čísla).

2.4.1.2 Interview

Úspěch nebo neúspěch interview záleží na analytických komunikačních schopnostech. Uživatel, kterému není analytik sympatický si pravděpodobně neudělá o budoucím systému dobrý obrázek bez toho, aby se vůbec o systém zajímal.

Analytik musí vytvářet dojem, že se dotazuje proto, aby řešil uživatelské problémy. Uživatel pak musí nabýt dojmu, že v tom může analytikovi pomoci. Pocit zainteresovanosti je důležitým faktorem úspěchu interview.

Před vlastním interview je nutná určitá příprava. K dispozici by měly být kopie dokumentů a formulářů na které může přijít řeč. Není nic horšího, než ztráta času hledáním patřičných dokumentů během vlastního interview. Analytik by měl

předem znát pozici dotazovaného a mít před schůzkou povolení od nadřízeného dotazovaného. Je vhodné nastítnit předem dotazovanému, nač se ho budeme ptát. Cílem interview není načapat dotazovaného, jak neví, ale zjistit od něj vše, co potřebujeme.

Interview se může konat jak v klientově kanceláři, analytikově kanceláři, nebo v neutrální místnosti jako je konferenční sál. V klientově kanceláři se dotazovaný cítí méně nervózně. Na druhou stranu může být interview přerušeno telefonáty a náhodnými návštěvami.

Tradiční interview je obvykle vedeno přímo mezi dotazovaným a analytikem. Pakliže je v místnosti více osob, většinou to vlastní interview pouze komplikuje.

Někdy se aplikuje metoda skupinového interview (Joint Application Development, podle IBM). Analytik se pak snaží moderovat skupinu tak, aby mohla specifikovat požadavky na systém. V případě dobrého týmového ducha v organizaci je možné dojít touto metodou k cílovým informacím dříve, než metodou individuálních interview.

Výsledky interview by měly být evaluovány hned vzápětí, neboť je vysledováno, že minimálně 50% informace se při delší odmlce vytratí.

2.4.2 Funkční analýza

Funkční model představuje dynamický pohled na systém. Vyjadřuje činnost systému tak, že si všímá míst kde je nutné transformovat data do jiné formy, a označuje je jako funkce. Existuje dvojí způsob jak hledět na data a na základě tohoto pohledu dělíme data na vstupní a výstupní. Jako standardním zápisem funkční analýzy je Diagram datových toků – Data Flow Diagram. Ten je většinou doplněn slovním popisem funkcí systému. Často se pro popis specifických funkcí používá pomocných nástrojů jako je např. rozhodovací tabulka.

Data Flow Diagram zobrazuje určitá data, jak procházejí určitou částí systému. Přes svůj název Diagram Datových toků je smyslem tohoto nástroje zobrazit procesy působící na tato data.

2.4.2.1 Elementy DFD

Ačkoli existuje několik variací DF diagramu, existují základní čtyři elementy objevující se ve všech.

Hraniční symbol (terminator) je člověk, nebo organizace postavená mimo systém, která je ale se systémem spojena

Terminator

datovým tokem (buď přijímá výstup, nebo poskytuje vstup). Jako analytika nás nezajímají procesy v rámci tohoto elementu, ale čistě data která poskytuje, nebo vyžaduje. Někdy se používá termín externí entita, ale v následujícím textu se pojem entita pojí s návrhem databáze, takže by bylo používání tohoto termínu matoucí.

Datová základna je „zásobárnou“ dat, která jsou v případě potřeby k dispozici. Fyzicky se může jednat o tabulku databáze stejně jako o tabulku spreadsheetu, nebo kartotéku.

Datová základna

Proces symbolizuje činnost, která musí být učiněna, aby byl splněn úkol, funkce. Data vystupující jsou zpravidla značně odlišná od dat do procesu vstupujících. V případě DFD vidíme, že se s daty cosi provedlo, nevíme/nedefinujeme ale jak.

Proces

Symbol toku dat je znázorněn šipkou a charakterizuje pohyb dat mezi procesy a z a do hraničních elementů.

2.4.3 Datové modelování

Datový model je vyjádřením statického pohledu na realitu. Zachycuje objekty, které nás zajímají, a vzájemné vztahy mezi nimi. Pro IS Hospodářské fakulty, v části 5.3.1 - Konceptuální model, rozvádím použití tzv. ERA diagramu (Entity-Relationship-Attribute). ERA diagram je model snažící se popsat základní tři skutečnosti:

- objekty, které nás zajímají (entity),
- vztahy existující mezi těmito entitami,
- podstatné vlastnosti (atributy) entit.

2.4.4 Entita

Entity představují něco, co je předmětem našeho zájmu. Je nutné diferencovat mezi typem entity a výskytem entity. Při modelování reality pracujeme s typy entit. Entita musí mít svoji definici, popis ze kterého lze vyčíst, jaký objekt entita reprezentuje.

2.4.5 Atribut

Atributy jsou podrobnosti, které nás v souvislosti s entitou (nebo i vztahem) zajímají. Atribut by měl mít nějakou hodnotu pro každý výskyt entity. Pakliže je zaručena jedinečnost atributu, tj. pro každý výskyt entity je atribut jiný (např. číslo indexu u studenta), mluvíme o identifikačním atributu. Dalším speciálním typem atributu je atribut opakovaný, kdy pro jeden atribut entity může existovat více hodnot (např. jazykové znalosti pedagoga).

2.4.6 Vztahy

Při definování vztahů mezi entitami není nejdůležitější najít veškeré možné vztahy, ale naopak vystihnout vztahy podstatné.

2.4.7 Datový slovník

Pojem Data dictionary je známý zejména ze systémů řízení báze dat (DBMS, viz kap. 4.4 na straně 28). Slouží k popisu struktury datové základny. Jedná se v podstatě o „encyklopedii systému“, kde jsou stručně popsány jednotlivé prvky systému. V rozsáhlejších projektech má správu DD většinou na starosti program, takže odpadá manuální údržba (o produktech CASE se zmíním v kapitole 2.9 na straně 22).

2.5 Návrh (design)

Na základě dat získaných v předchozí fázi být musí hned zpočátku rozhodnuto o požadavcích na zařízení. Hned od počátku musí být k dispozici jak HW, tak Software nutný pro vývoj systému.

Funkční diagramy z analytické fáze je třeba transformovat do podoby hierarchických schémat návrhové fáze. Značná část úsilí je směřována na uživatelské rozhraní (UI - user interface).

Je nutné též specifikovat konkrétní nároky na lidské zdroje nutné k návrhu, implementaci a údržbě systému.

Stanovené požadavky na funkce systému by měly být přetransformovány na konkrétní programy. Analytik se rozhoduje jak program strukturovat, navrhuje uživatelské rozhraní a hierarchii nebo formu jak budou programy strukturovány. V této fázi je nutné brát v potaz i bezpečnostní faktory jako je zabezpečení proti vniknutí a vůbec veškerá přístupová politika.

Do návrhu uživatelského rozhraní patří návrh jednak vstupních formulářů, tak formátování výstupu. Při tvorbě vstupních formulářů je třeba snažit se minimalizovat možnost vstupu nekorektních dat.

2.6 Realizační fáze

V realizační fázi jsou psány a testovány vlastní programy, je připravována dokumentace a výukový materiál, tak aby byl k dispozici pro fázi přechodu na nový systém. Výsledkem je naprogramovaný a otestovaný systém, připravený k přechodu na ostrý provoz.

V případě, že se nejedná o vlastní produkt, je nutné zakoupený balík upravit pro potřeby organizace.

2.7 Přechod na nový systém

Tato fáze spočívá v přechodu organizace od starého k novému systému. Systém se plní „ostrými“ daty. Ve většině případů je možná elektronická konverze dat ze starého do nového systému, ale zřídka se setkáváme se situací, kdy není nutný manuální vstup.

Konverze může probíhat průběžně. Z bezpečnostních důvodů je často používán postup paralelního běhu obou systémů.

2.8 Údržba

Tato dlouhodobá fáze probíhá za účelem udržení systému ve funkčním stavu. Udává se, že velké organizace směřují 50-70% úsilí programátorů právě na údržbu existujících systémů.

Existují dva hlavní důvody proč je tato fáze tak důležitá: jednak dochází během fungování k nepředvídatelným chybám a jednak mohou a nastávají změny v reálném prostředí organizace, které musejí být reflektovány i v IS.

Analytik bývá na vzniklý problém upozorněn uživatelem. Ten připraví model odstranění tohoto problému a konzultuje tento zásah do systému s vedením organizace. Po odsouhlasení vedením je postup postoupen programátorovi, který patřičné změny v systému realizuje. Poté je nutné řádně systém otestovat a další fungování systému již probíhá změněným postupem.

V praxi se většinou nesetkáváme s tím, že by jednotlivé fáze na sebe přímo navazovaly. Jednotlivé kroky se často prolínají. Analytická část není ještě dokončena a už se začíná pracovat na návrhu.

Pro ilustraci fází životního cyklu informačního systému nabízí [3] porovnání s výstavbou nové výrobní haly (tabulka 1).

tabulka 1 - porovnání životního cyklu IS a podnikového závodu

Stavba nového závodu	Tvorba nového IS
Definování problému: Vedení podniku zjišťuje, že prostory současného závodu jsou nedostačující.	Vedení zjišťuje, že současný IS pro potřeby podniku nedostačuje.
Studie proveditelnosti: Architekt provede předběžnou studii aby zjistil hrubé náklady a problémové oblasti	Analytik provede předběžnou studii aby zjistil hrubé náklady a problémové oblasti
Analýza: Architekt prostuduje dokumentaci a výkresy současné budovy závodu a hledá problémy a požadavky na novou budovu.	Analytik hledá problémy současného a požadavky na nový informační systém
Návrh/Design: Architekt vytváří výkresy nové budovy.	Analytik vytváří modely budoucího IS.
Vývoj: Stavební firma pod vedením architekta provádí stavební práce. Nad stavbou bdí stavební inspekce.	Programátoři vytváří, pod vedením analytika, nový IS
Přechod: Technika a personál se stěhuje do nové budovy a rozjíždí se výroba	Data, personál a postupy přechází na nový systém
Údržba: Nový závod je udržován spec. útvarem až do doby, kdy je rozhodnuto o jeho zastaralém stavu a celý životní cyklus začíná znovu.	IS musí být upravován a udržován v provozuschopném stavu, rozšiřován analytiky a programátory/administrátory až do doby kdy je rozhodnuto o jeho zastaralosti a přechodu na nový. Začíná životní cyklus nového IS.

2.9 Systémy CASE

CASE, neboli Computer Aided Software/System Engineering jsou softwarové produkty sloužící k automatizaci procesu tvorby informačních systémů. Stejně jako u svého protějšku CAD (Computer Aided Design) znamená obrovskou úsporu času oproti klasickým ručním metodám.

Podle funkčního záběru se CASE rozděluje do tří skupin:

2.9.1 Upper/Front-End CASE

Se používají zejména ve fázi Analýzy a návrhu systému. Součástí bývá většinou datový slovník, nástroje pro tvorbu diagramů datových toků, datových modelů a strukturních diagramů. Obsahuje silné prostředky pro formální prezentaci výsledků.

2.9.2 Middle CASE

Umožňují analýzy a návrhy datových struktur, analýzy a návrhy, případně i optimalizace funkčních struktur, prototyping, obsahuje prostředky pro tvorbu dokumentace (systémová příručka).

2.9.3 Lower CASE

Přestože funkční rozpětí může být značné, jádro CASE je v podstatě vždy stejné. Je to databáze, do které se ukládají výsledky projekčních kroků a která se nazývá Data Dictionary, Repository nebo systémová encyklopedie. Tuto databázi CASE automaticky udržuje v konzistentním stavu. Všechna pravidla, která lze v rámci podporované metodologie algoritmizovat, CASE stále aplikuje na uložené výsledky. Jakmile chceme provést něco, co odporuje těmto pravidlům, CASE nám to nedovolí a zabrání tak řadě formálních nebo logických chyb.

V oblasti datové analýzy pomáhají CASE systémy při konstrukci konceptuálního modelu, většinou ve formě ERA diagramu, normalizaci datových struktur a transformaci do logické struktury dat, nejčastěji do relační struktury. Ve fázi funkční analýzy podporují hierarchickou dekompozici systému na diskrétní části, vyjádření toku dat mezi částmi systému a mezi systémem a jeho okolím. Zároveň umožňují formulovat podmínky vykonávání jednotlivých činností.

3. INTRANET

Platformová nezávislost a multimediální charakter služby WWW ji činí ideálním prostředkem prezentace dat. Množství organizací, zejména z komerčního sektoru začalo pocítovat potřebu využít služby WWW pro prezentaci dat uložených v některé podnikové databázi.

Pro podnikové informační systémy založené na přístupu k podnikovým databázím přes WWW se ustálil termín Intranet. Vlastní připojení ke globální síti Internet není u Intranetu podmínkou.

Hlavní odlišností od stávajících systémů je z pohledu uživatele stejné rozhraní, nezávislé na klientově platformě. Jelikož právě u velkých organizací existuje rozmanité spektrum různě zastaralých pracovních stanic, toto řešení ideálním způsobem integruje všechny tyto platformy.

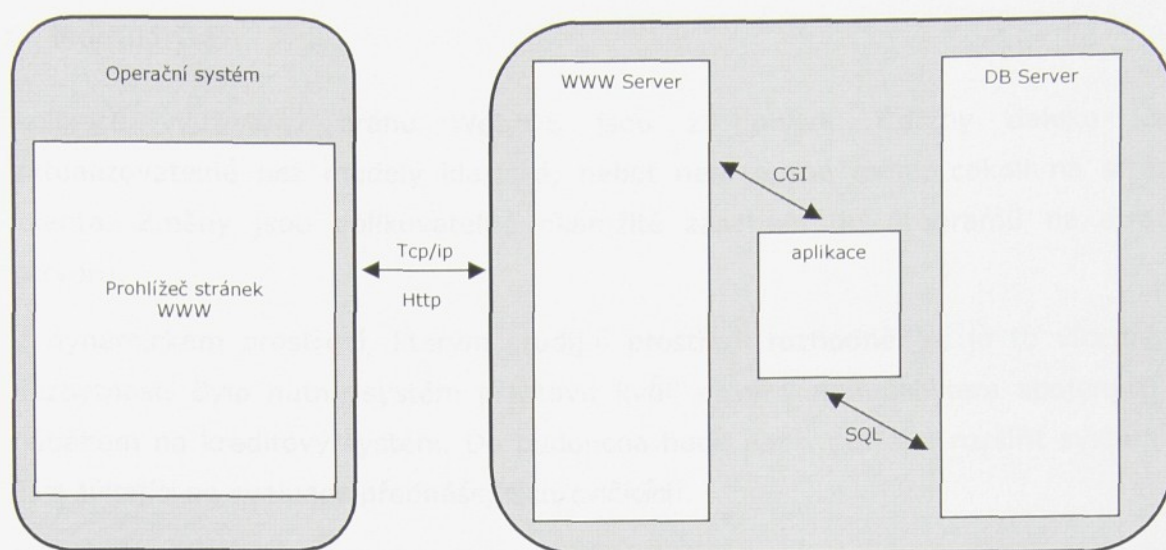
Základním předpokladem vzniku fenoménu Intranet byla již zmíněná služba WWW založená na přenosu hypertextových dokumentů a dalších multimediálních dat. I když klient formátuje zdrojový dokument zaslaný serverem, jedná se o dokument statický. Pravda, existují aktivní prvky na straně prohlížeče jako je JavaScript, VBScript, Java applety apod., ale v případě intranetových aplikací je dynamičnost většinou zabezpečena na straně serveru.

Pro dynamicky generované dokumenty se začal používat mechanismus CGI – Common Gateway Interface. Tento mechanismus umožňoval WWW serveru předat data, která dostal od klienta, například z vyplněného formuláře, externímu programu, který pak zase přes CGI vrací serveru výstup a ten jej zformátuje standardní formou pro WWW klienta.

Z pohledu klienta tedy není rozdíl mezi tím, zda pracuje se statickým dokumentem, nebo dokumentem generovaným. Externí program volaný přes CGI nemusí být uzpůsoben podmínkám klientského počítače, neboť jemu posílá výstup WWW server a tato komunikace probíhá na základě standardního protokolu HTTP v HTML formátu. Program musí být uzpůsoben prostředí ve kterém běží WWW server a musí umět komunikovat přes rozhraní CGI.

Obrázek 2 ukazuje klasické řešení intranetového systému pomocí CGI brány.

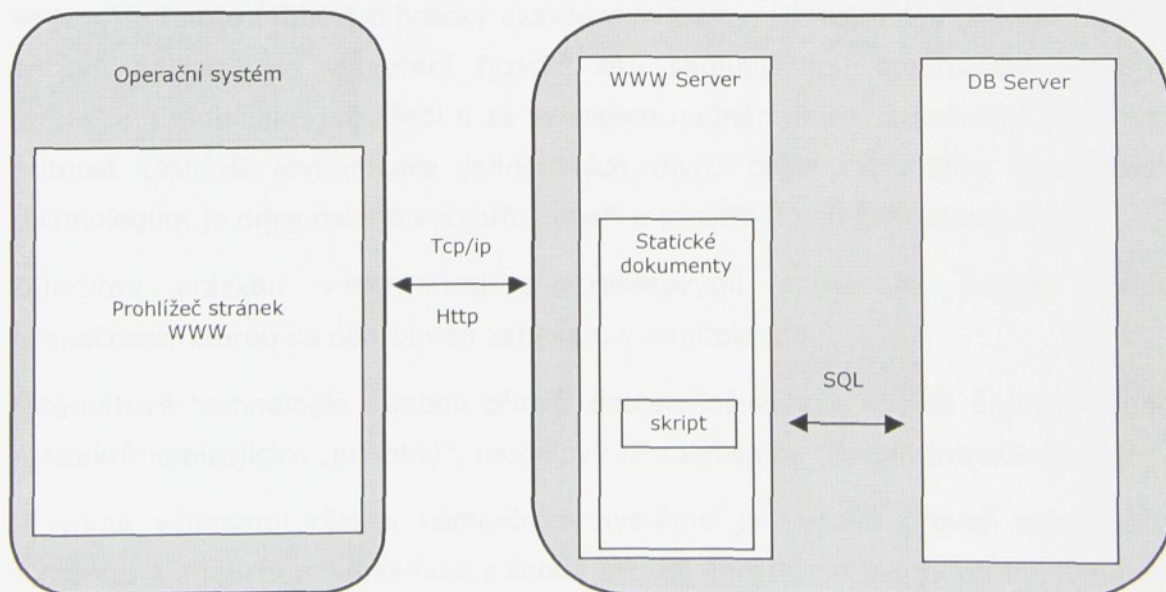
Obrázek 2 – skript na straně serveru



obrázek 2 - CGI rozhraní

Již na první pohled je ale vidět určitá neefektivnost toho řešení. WWW server pomocí rozhraní CGI (common gateway interface) funguje pouze jako zprostředkovatel. Dotazování na databázový server je činěno externím programem který pak zase výstup posílá přes rozhraní zpět WWW serveru.

Daleko efektivnější, tedy jednak rychlejší, tak méně náročné na systémové prostředky web serveru je použití skriptů na straně serveru (tzv. server-side scripting). V běžném HTML dokumentu je speciální kód, který se před odesláním klientovi vykoná a zabezpečí konektivitu na DB server.



obrázek 3 - skript na straně serveru

3.1 Modularita

Aplikace využívající bránu Web/DB jsou z pohledu údržby daleko lépe aktualizovatelné než modely klasické, neboť není nutné měnit cokoli na straně klienta. Změny jsou aplikovatelné okamžitě zásahem do programů na straně serveru.

V dynamickém prostředí, kterým studijní prostředí rozhodně je, je to víceméně nezbytnost. Bylo nutné systém přestavit kvůli novým skutečnostem spojených s náběhem na kreditový systém. Do budoucna bude např. potřeba rozšířit systém o část týkající se evaluace přednášejících/cvičících.

Nároky kladené na klienta jsou minimální, a správně napsaná aplikace by měla být použitelná i na starších strojích používajících např. textového klienta, který pro svojí funkčnost nepotřebuje náročné grafické prostředí.

Díky jednotnému rozhraní na všech klientských prohlížečích se tímto řešením minimalizují nároky na školení uživatelů. Uživatel, který již umí pracovat s prohlížečem WWW a formuláři již v podstatě umí systém ovládat.

3.2 Extranet

Termín extranet se v současné době začíná používat v souvislosti s organizacemi, které vytvářejí své Intranetové aplikace v otevřeném prostředí Internetu. Nejedná se tedy v tomto případě o fyzicky uzavřené systémy. Hlavním důvodem je zejména snížení nákladů na vytvoření fyzické sítě spojující její organizační jednotky. V dnešním globálním prostředí a za existence nadnárodních společností stále roste nutnost těsnější komunikace jednotlivých divizí organizace. Díky kryptovacím technologiím je dnes možné vytvořit „tunel“ v prostředí veřejné datové sítě.

Důležitou otázkou v souvislosti s extranetovými aplikacemi ovšem zůstává bezpečnost, kterou se podrobněji zabývám v kapitole 4.5.

Internetové technologie s sebou přináší dostatečně výhod, aby se objevilo mnoho produktů nabízejících „prázdný“, modelový IS založen na filosofii Intra/Extranetu.

Hlavními výhodami těchto komerčních systémů je vysoká úroveň zabezpečení systému, ale jejich univerzálnost s sebou přináší nemalé problémy při implementaci v konkrétních podmínkách.

Jedny z nejnáročnějších Internetových aplikací na bezpečnost jsou systémy elektronického bankovníctví. I u nás již existují funkční systémy tohoto druhu – uveďme např. projekt Expandia banky.

4. TECHNICKÉ PROSTŘEDKY

4.1 HTML

HTML neboli hypertext markup language je standard definující strukturu hypertextového dokumentu používaného v prostředí WWW. Pomocí speciálních značek, uzavřených v „<>“ závorkách, definuje speciální části dokumentu. Dokument je serverem posílán klientskému prohlížeči jako obyčejný text a teprve prohlížeč tento dokument zformátuje do finální podoby.

4.2 PHP³

PHP neboli professional hypertext preprocessor je velice silný skriptový jazyk. Základ jazyka byl vytvořen v roce 1994. Jeho autor, Rasmus Lerdorf ho napsal, aby mohl vnést prvek dynamičnosti do svých domácích stránek. Začátkem roku 1995 byly zdrojové kódy nabídnuty veřejně k dispozici, jak je uvedeno v General Public licenci, pod kterou je PHP distribuováno. V té době dostal jazyk název Personal Home Page Tools. Tou dobou se jazyk hodil na často potřebné úkoly jako vytvoření počítadla, nebo knihy návštěv. Teprve v polovině roku 1995 byl do programu zapracován kód pro spolupráci s html formuláři a produkt přejmenován na PHP/FI (Form Interpreter). V této době Rasmus Lerdorf zakomponoval do jádra PHP tak podporu tehdy populárního DBMS mSQL (mini SQL). Díky zvolené licenční politice začalo na tvorbě PHP spolupracovat větší množství autorů a společně s vývojem jazyka přibývalo uživatelů.

Zatímco koncem roku 1996 bylo PHP/FI používáno zhruba na 15 000 web serverů, v polovině roku 1997 už toto číslo vzrostlo na 50 000. V této době také zaznamenáváme přerod celého projektu. Jádrem programu – překladače se od této chvíle chopil Zeev Suraski a Andi Gutmans a PHP dospělo do verze 3. PHP3 již má se svým předchůdcem společnou pouze filosofii. Vesměs nic z původního kódu již v novém PHP není zachováno.

PHP dnes funguje jako základní stavební kámen mnoha komerčních řešení firemních intranetů, jako např. C2 Stronold Server a na základě dat od společnosti NetCraft se počítá se zhruba 150 000 web servery používajících PHP. (Toto číslo je dnes

³ Date, C.J, White, C.J.: PHP3 manual. <http://www.php.net>

pravděpodobně zase větší). Pro srovnání je to více, než kolik prodala společnost Netscape svého hlavního produktu – Netscape Enterprise Server.

4.3 SQL⁴

V současnosti je nejrozšířenějším relačně úplným jazykem SQL. SQL (Structured Query Language) je v současnosti implementován téměř do každého DBMS.

Jazyk začala vyvíjet společnost IBM již na začátku 70. Let. Cílem tvůrců SQL bylo vyvinout takový nástroj pro koncové uživatele, který by jim umožňoval vybírat data z databáze přesně podle jejich požadavků a byl přitom co nejjednodušší. První část záměru se podařilo realizovat celkem úspěšně. SQL je mimořádně silný dotazovací prostředek. Kromě dotazování můžeme s jeho pomocí definovat data, provádět aktualizace, a v některých implementacích i vytvářet indexy.

Dnes je zřejmé, že původní záměr autorů SQL nebyl reálný. SQL je pro koncového uživatele příliš složitý. I přes to se standard SQL prosadil a představuje jeden ze silných standardů. Prosadil se však nikoli díky jednoduchosti pro koncového uživatele, nýbrž pro svoji nenahraditelnou roli mostu mezi různými systémy. SQL funguje jako standardní rozhraní. Bohužel se můžeme setkat s několika mutacemi standardu, což trochu kalí funkci SQL jako ideální rozhraní.

SQL je možné použít jako dotazovací jazyk pro interaktivní zadávání dotazů v terminálu, nebo jako prostředek pro komunikaci aplikačního programu s databází. V druhém případě jsou příkazy SQL vloženy přímo do zdrojového kódu tradičního programovacího jazyka.

Obecně lze rozdělit typy příkazů do dvou skupin:

- příkazy pro definici dat (CREATE, ALTER, DROP),
- příkazy pro manipulaci s daty (DELETE, INSERT, SELECT, UPDATE).

4.4 Databázový systém MySQL⁵

MySQL je víceuživatelský SQL server. Kromě vlastního serveru mysqld, obsahuje balík i mnoho klientských aplikací a knihoven.

⁴ Straka, M.: Vývoj databázových aplikací. 1.vyd. Praha 1992, Grada, str. 80

⁵ MySQL reference manual, <http://www.mysql.org>

Hlavním cílem autorů bylo vytvořit rychlý, robustní databázový systém s jednoduchou administrací. Celý systém byl původně vytvořen pro interní účely švédské firmy TcX. Firma potřebovala rychlý databázový systém, který by uměl pracovat s rozsáhlými tabulkami. Ve firmě TcX používají MySQL od roku 1996 pro správu více než čtyřiceti databází obsahujících více než 10 000 tabulek z nichž více než 500 jich má přes 7 miliónů záznamů. Dohromady to činí kolem sto gigabajtů pro organizaci vitálně důležitých dat.

I když je MySQL stále ve vývoji, už dnes nabízí dostatečně bohatý a užitečný výběr funkcí.

Původně autoři chtěli použít pro své databáze předchůdce dnešního systému MySQL – produkt mSQL (mini-SQL), ale brzy zjistili, že nedosahuje požadovaného výkonu. Výsledkem práce na MySQL byl systém, se stejným aplikačním rozhraním (API) jako mSQL.

Mimo jiné nese MySQL tyto hlavní rysy:

- Multithreaded – používá vlákna (threading) jádra systému (kernel), takže umí využít více procesorů.
- API pro C, C++, Javu, Perl, Python, TCL a další.
- Pracuje na mnoha operačních systémech.
- Velmi rychlé pojení tabulek (one-sweep multi-joins).
- SQL funkce jsou zabezpečeny přes vysoce optimalizovanou třídivou knihovnu.
- Plná podpora pro SQL funkce „GROUP BY“ a „ORDER BY“. Podpora skupinových funkcí „COUNT()“, „AVG()“, „STD()“, „SUM()“, „MAX()“ a „MIN()“.
- Podpora pro ANSI a ODBC syntaxi SQL.
- Možnost použít tabulky z různých databází v jediném dotazu.
- Konektivita přes ODBC 2.5 (včetně zdrojových kódů). Takto je možné používat například rozhraní MS Access pro práci s MySQL databází.
- Je možné vytvořit až 16 indexů v jedné tabulce.
- Záznamy s pevnou nebo proměnlivou velikostí proměnné.
- Implicitní hodnota pro jakýkoliv sloupec.
- Plná podpora pro ISO Latin2, včetně českého řazení.

- Tvorba aliasů na tabulky a sloupce podle standardu SQL92.
- Při použití „INSERT“, „DELETE“, „REPLACE“, „UPDATE“ vrací informaci o tom, kolik záznamů bylo upraveno.
- Lokalizace prostředí.
- Připojení k serveru pomocí TCP/IP.
- Klienty/podporu pro MySQL nalezneme v mnoha produktech (PHP, Roxen Chllanger Web Server, IHTML, FastFlow NDC, MetaHTML, HeiTML ...).
- Produkt je distribuován zdarma formou zdrojového kódu, a je obsažen přímo v mnoha distribucích operačního systému Linux. U verze pro operační systém Windows je nutné zakoupit licenci. Při prodeji produktu nebo služby obsahující MySQL třetí osobě, je též nutné zakoupit licenci.
- Kód klienta MySQL je distribuován podle podmínek GNU (<http://www.gnu.org>).
- Neexistuje žádné omezení v počtu klientů připojených současně k serveru, ani v počtu současně běžících serverů na jednom stroji, jako je to běžné u komerčních produktů.

Databázové systémy jak je známe z komerčního světa bývají kompletními balíky se sofistikovanými nástroji pro tvorbu a údržbu databáze. MySQL působí v tomto ohledu dost chudě. Důvodem je vlastní koncepce tohoto produktu. Cílem tvůrců bylo vytvoření robustního DBMS, ve kterém není implementováno nic, co by omezilo rychlost. Na druhou stranu je díky otevřenosti zdrojového kódu k dispozici mnoho klientů MySQL, které umožňují administraci databází vizuálně (např. MyAdmin, kmysql).

4.5 Bezpečnost⁶

Ačkoli si tato práce nedělá ambice rozvádět problematiku bezpečnosti do detailů, považují za nutné se alespoň zmínit o základních principech bezpečnosti v prostředí Internetu.

⁶ CERN: Security for a public web-site: <http://www.cert.org/security-improvement/modules/m02.html>

Na úvod této kapitoly je nutné konstatovat, že v prostředí Internetu neexistuje 100% bezpečnost. Globální povaha této sítě způsobuje, že útoky na bezpečnost různých sítí v rámci Internetu je velice častým jevem. V případě sítí lokálnějšího charakteru existovala u škůdce zábrana v podobě rizika, že v případě odhalení budou proti němu uskutečněna okamžitá opatření. V případě útoku z druhé polokoule je toto fyzické riziko minimální a zábrany u „hackera“ jsou téměř nulové.

V rámci bezpečnostní politiky chceme zabezpečit následující případy:

Utajení - data, která nechceme zpřístupnit veřejnosti.

Integrita - nechceme umožnit neautorizovaným osobám upravovat data.

Přístupnost - naopak osobám autorizovaným umožnit s daty manipulovat.

V souvislosti s podnikovým extranetem existuje několik hypotetických bezpečnostních „děr“:

- nespolehlivost uživatele,
- www (http) server,
- databázový server.

4.5.1 Uživatel

Ačkoli je většina bezpečnostních opatření směřována externě, v mnoha případech je porušení bezpečnosti zaviněno, způsobeno interním zaměstnancem. Existuje ale mnoho represivních opatření, zamezujících budoucí opakování této situace (rozvázání pracovního poměru atd.).

4.5.2 WWW server

Bezpečnostní mezera ve vztahu k serveru (kombinace hardware+software zabezpečující provoz webových aplikací) může nastat v souvislosti s následujícími okolnostmi:

- špatná konfigurace serveru vedoucí k umožnění přístupu k neveřejným datům.
Tato data mohou být následujícího charakteru:
 - Konkrétní data týkající se organizace,
 - informace o interní síti, topologii sítě a další informace zneužitelné v jiném způsobu útoku,
 - informace o tom kdo získal/čte určité dokumenty ze serveru.

- Cílem útoku je vlastní počítač, který nabízí webovou službu.
 - Narušitel pak může změnit data, které hodlá administrátor publikovat na WWW.
 - Narušitel získá možnost nabourat se do vnitřní sítě.
 - Narušitel může použít počítač k dalším útokům na cizí server, a může tím ohrozit pověst dané organizace.

Dodržení bezpečnosti je realizováno ve třech krocích:

- 1) výběr software serveru,
- 2) konfigurace software,
- 3) administrace serveru.

Nezbytnou součástí bezpečnostní politiky je i obeznámení uživatelů a definování odpovědnosti.

tabulka 2 - doporučené bezpečnostní praktiky

Oblast	Doporučený postup
Výběr software	- Explicitně vyjádřit požadavky na bezpečnost
Konfigurace	- Izolovat server od interního systému organizace - Na stejném stroji nabízet jen nutné síťové služby - Velkou pozornost věnovat konfiguraci WWW serveru - Speciální důraz klást na programy/skripty vykonávající něco externě
Provoz serveru	- Administrace bezpečným způsobem - Hledání podezřelých souborů, adresářů - Sledování systémových a síťových log souborů

4.5.2.1 Explicitní vyjádření požadavků na bezpečnost

Kromě výkonnostních kritérií, která obvykle znamenají množství současně zpracovávaných dotazů, objem dat odeslaných serverem za časovou jednotku, je nutné brát v potaz i následující bezpečnostní kritéria, která ve svém důsledku mohou znamenat snížení nákladů spojených se znovuzprovozněním serveru v případě průniku do systému:

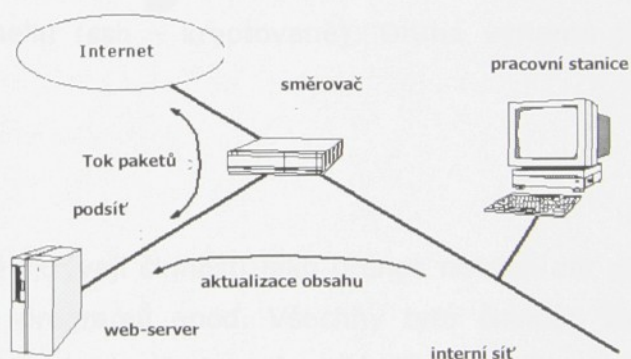
- Server software je imunní k již známým způsobům průniku.

- Možnost chráněného přístupu pro administrátory.
- Schopnost zakázat přístup pro dokumenty, které nejsou míněny k veřejnému publikování.
- Možnost vypnout některé síťové služby, zabudované do server-software nebo operačního systému.
- Schopnost konfigurovat externí programy (CGI skripty/serverové pluginy).
- Možnost vytvářet historii určitých aktivit (log) pro zpětné zjišťování narušení bezpečnosti.

4.5.2.2 Izolovat server od interního systému organizace

Pakliže je to možné, je vhodné fyzicky umístit webový server na speciální podsít, aby nebylo možné „odchytávat“ provoz interní sítě. Veřejný webový server je server určen pro širokou veřejnost. To znamená že k počítači má přístup kdokoli z okolního světa. Jakkoli je zabezpečení dokonalé, vždy existuje nějaký nový způsob, nové zranitelné místo, které může umožnit přístup ke stroji, na kterém webový server běží. Z tohoto důvodu je třeba zabezpečit následující stavy:

- narušitel je schopen na stanici „odchytávat“ pakety interní sítě. V těchto paketech se můžou nacházet autorizační data, nebo konkrétní obchodní, osobní informace.
- Narušitel je schopen se dostat na další interní stanice



obrázek 4 - odizolování WWW serveru od interní sítě

Z těchto důvodů je vhodné web server izolovat od interní sítě. Docílit toho lze pomocí speciálních bran/firewallů.

Tyto brány zabezpečí filtrování datagramů jak je uvedeno na obrázku 4.

4.5.2.3 Na stejném stroji nabízet jen nutné síťové služby

Je dnes již trendem, že jediný stroj zabezpečuje hned celou řadu síťových služeb. Z pohledu bezpečnosti je tato tendence nežádoucí. Důvody jsou několiké:

- Jiné služby mohou umožnit zásah do webových dat, nebo je odstranit.
- Jiné služby mohou být v kompetenci jiných administrátorů a může dojít ke kolidujícím zásahům.
- Různé služby mají různé nároky na hardware
- Méně služeb znamená méně historických souborů (log files), a tedy snazší orientaci v nich.

Nejběžnějším postupem je postup nazývaný „zakaž, pak povol“ (deny first, then allow). Tato metoda spočívá v prvotním zákazu všech služeb a jednotlivému povolení potřebných služeb. Tímto způsob je možné se vyvarovat opominutí té či oné nepoužívané služby.

Pakliže existuje více způsobů jak službu poskytnout, je vhodnější ta bezpečnější. Například vzdálená administrace je v Unixových systémech možná pomocí tzv. remote shellu (rsh), nebo secure shellu (ssh - kryptovaně). Druhá varianta je bezesporu bezpečnější.

4.5.2.4 Administrace serveru

Pod pojmem administrace serveru se skrývají činnosti jako přenos nových dat na server, instalace a úprava externích programů apod. Všechny tyto činnosti lze provádět jak přímo na konzoli serveru, tak prostřednictvím sítě. V obou případech je nutné, aby administrátor neumožnil třetí osobě proniknout do stanice.

Ačkoli může být server z hlediska bezpečnosti v pořádku, během administrace se může stát zranitelný. To platí zejména v případě vzdálené administrace.

V případě, že se správce rozhodne pro vzdálenou administraci, je nutné dbát výjimečný důraz na bezpečnost.

- Pro autorizaci přístupu není vhodné používat prostý text.
- Pro vzdálenou administraci je dobré definovat jen jednu stanici (hostitelský počítač - host). Autorizace by ale neměla záviset pouze na adrese stanice (pomocí IP adresy, nebo DNS jména). Tuto informaci lze „zfalšovat“.

Během aktualizace je možné server vyřadit z provozu (server odpovídá klientům hláškou „Service unavailable“).

V případě, že kontrolujete historii přístupů (log file), používejte bezpečné připojení.

4.5.2.5 Externí programy/skripty

Existuje mnoho případů, kdy byla chyba v nějakém veřejně distribuovaném CGI programu zneužita k útoku na server.

V případě že sami vytváříme takové skripty, takže odpadá nutnost kontrolovat důvěryhodnost zdroje, na co je třeba se z pohledu bezpečnosti zaměřit?

Musíme si položit dvě hlavní otázky:

- je možné použít již existující skript k něčemu, k čemu nebyl vytvořen, například podstrčením nekorektního vstupu?
- Je pro narušitele možné umístit na server vlastní skript a spustit ho?

Druhá varianta lze korektním nastavením serveru eliminovat úplně. První způsob narušení bezpečnosti je ovšem daleko hůře ošetřitelný. Prvním krokem k jeho eliminaci je uvědomění si možností, které útočník má. Kritickým se program stává v případě použití uživatelského vstupu. Je tedy nutné jeho vstup nejprve kontrolovat a teprve korektní vstup postoupit dalšímu zpracování.

5. INFORMAČNÍ SYSTÉM HOSPODÁŘSKÉ FAKULTY

5.1 Definování problému

Od roku 1993 spolu komunikují studenti a fakulta, kromě klasické formy nástěnek a studijního oddělení, přes počítačový systém založený na prostředku lokální sítě.

Původní systém sloužil k jednostrannému informování studentů vedením fakulty, a jednotlivých kateder. Student měl možnost získat informace o vyučovaných předmětech a to v následujícím rozsahu:

- **Obecné informace o fakultě** – Sídlo, vznik, vedení fakulty, studijní programy, počty uchazečů o studium, systém studia a další všeobecné informace.
- **Studijní plán** – pro jednotlivé studijní programy byla k dispozici tabulka předmětů vyučovaných určitý semestr s rozpisem počtu přednášek, cvičení týdně, a způsobu klasifikace. V době fungování systému byl pevně daný studijní program pro jednotlivé semestry.
- **Rozvrhy** – tato část byla zpočátku nejpoužívanější částí celého systému, ale později docházelo ke stále větším skluzům v aktualizaci oproti nástěnkám v budově HF, takže význam této části IS stále klesal.
- **Sylabus předmětu** – zde mohli studenti nalézt informace o struktuře přednášek/cvičení, návaznosti na další předměty, přednášející, cvičící apod.
- **Aktuální novinky** – pro účely rychlého informování studentů o důležitých událostech týkajících se studia na HF, byl vytvořen systém elektronických nástěnek. Každý student, byl informován o nových zprávách na virtuálních nástěnkách po přihlášení do lokální sítě Liane.

Ve své bakalářské práci jsem se pokusil zvětšit dosah této jednostranné komunikace použitím internetové služby WWW. Pokusím se rozebrat základní problémy spojené s používáním obou systémů.

- 1) Základním problémem zůstává aktuálnost dat, která je v obou zmíněných případech minimální. Důvodem je fakt, že v obou případech leží odpovědnost za aktualizaci poskytovaných dat na jediné osobě.
- 2) Registrace volitelných předmětů. Díky novému kreditovému systému studia je určitým způsobem porušen zavedený řád studia po semestrech s pevným

studijním plánem. Při velkém počtu studentů již nebylo únosné pokračovat ve stávající registraci jednotlivých zájemců o předmět přímo u vyučujícího.

- 3) Doposud se rozvrhy učeben a studijních kruhů publikovali na nástěnkách v budově Hospodářské fakulty. Elektronická podoba rozvrhů se v systému objevovala s velikým zpožděním. Studenti byli tedy závislí na rozvrzích vyvěšených na nástěnkách. Nejenom že tento způsob nebyl dostatečně rychlý, zároveň nesplňoval požadavek studentů získat rychle kopii tohoto dokumentu.
- 4) Pro účely porovnání úrovně přednášek a cvičení mezi různými ekonomickými fakultami vysokých škol v ČR byl vypracován požadavek na evaluační systém práce pedagogů. Tato část systému je zatím v přípravě a v této práci se jí zatím nezabývám.
- 5) Již delší dobu funguje na univerzitní půdě elektronický systém zabezpečující funkci telefonního seznamu. Doposud ale neexistuje přijatelné uživatelské rozhraní.
- 6) Pro účely fakulty a eventuálních uchazečů o studium by měly být k dispozici informace o absolventech studia na HF a s tím spojený systém administrace.

5.2 Studium proveditelnosti

Základní poskytovanou informací jsou informace o předmětech vyučovaných na fakultě. Jednak je to studijní plán, popisující průřezově celé studium programu, tedy jaké předměty musí student absolvovat. Dále je požadován podrobnější popis předmětu včetně sylabu semestru.

Hlavní důraz by měl být kladen na aktuálnost nabízených dat. U tak velké organizace není únosné, aby aktualizaci celého systému zabezpečovala pouze jediná osoba, jako tomu bylo v minulosti. Je tedy nutné vytvořit systém, který by umožnil osobám, které sice mají k dispozici potřebná data pro aktualizaci, ale není pro ně přístupný současný způsob aktualizace, tato data do systému vkládat. Kromě uživatelského rozhraní bylo nutné se zaměřit též na bezpečnostní stránku věci, neboť WWW služby jsou službami veřejnými a aktualizace musí probíhat alespoň částečně zabezpečeně.

Problém ovšem spočívá v tom, že osoby, které mají k dispozici patřičná data, nemohou (nemají čas, neumí) se stávajícím systémem přímo pracovat. Předávání dat neelektronickou formou další osobě, jak tomu v minulosti bylo, je ovšem silně

neefektivní a v podstatě nefunkční. Musí být tedy nalezeno řešení, které umožní i laikovi vkládat-aktualizovat data, které ovšem bude patřičně zabezpečeno proti neautorizovanému vstupu.

Uživatelské rozhraní služby World Wide Web je již dnes rozšířené natolik, že si troufám ho označit jako ideálním rozhraním komunikace mezi uživatelem a bází dat systému. Problematikou použití služby WWW v prostředí Intranetu jsem se již zabýval v kapitole 3 - Intranet.

V případě použití služby WWW jako uživatelského rozhraní pro podnikovou databázi se nejedná o žádnou žhavou novinku. Možná proto je v současné době k dispozici nespočet komerčních i nekomerčních řešení.

Pro Hospodářskou fakultu byla ale vybrána kombinace nekomerčních produktů. Rád bych toto rozhodnutí okomentoval.

Hlavní výhodou doporučeného řešení je jeho cena, ale také jistota neustálého vývoje produktu. GPL (General Public Licence) mimo jiné zaručuje, že program bude distribuován včetně zdrojového kódu. Za programy distribuované na základě této licence není možné žádat finanční odměnu (pouze za služby spojené s distribucí). Do budoucna je tedy zaručeno, že pakliže bude někdo, kdo produkt bude využívat, má možnost sám produkt vylepšovat podle svých potřeb. (Nebo si další vývoj o někoho objednat). Mnoho komerčních firem pochopilo výhodu poskytovat svůj produkt pod speciální licenci s odkazem na GPL. Např. norská firma Trolltech distribuuje své produkty na základě vlastní EUL (end-user-license) smlouvy, ale v případě, že upustí od dalšího vývoje, jako se to v komerčním světě často stává – firma např. ukončí svoji činnost, přejde celý zdrojový kód pod GPL. Takto získává firma se svým produktem konkurenční výhodu spočívající v delší životnosti produktu a motivuje uživatele k zakoupení produktu s jistou budoucností. I to svědčí o jiných výhodách „free software“, než pouze nízké ceně.

Při nákupu komerčního produktu získává zákazník kromě vlastního produktu též tištěnou dokumentaci a technickou podporu. U rozsáhlejších produktů si podniky objednávají služby konzultačních firem, které pomohou s náročnou fází implementace systému.

Co se týče „free“ produktů, dokumentace je na různé úrovni, ale za technickou podporu je nutné platit stejně jako u komerčních produktů.

Všeobecně se ale dá tvrdit, že použití nekomerčního řešení znamená zpravidla více úsilí při implementaci, než u lépe zdokumentovaných a „dopilovaných“ komerčních

ekvivalentů. Jen zřídka ale investice do studia produktu, školení vlastního zaměstnance, proškoleného odborníka, převyšuje cenu komerčního produktu který ale sám může vyžadovat další investice spojené s konkrétní implementací.

V případě použití databázového systému MySQL, nebude zapotřebí nikoho školit, protože jak na katedře informačních technologií, tak katedře informatiky je dostatek personálu znalého tohoto produktu natolik, aby se mohl věnovat jeho administraci. Totéž se dá říct o produktech PHP a Apache.

Z finančního pohledu tedy jedinou znatelnou položkou je mzda analytika, programátora a posléze administrátora celého systému.

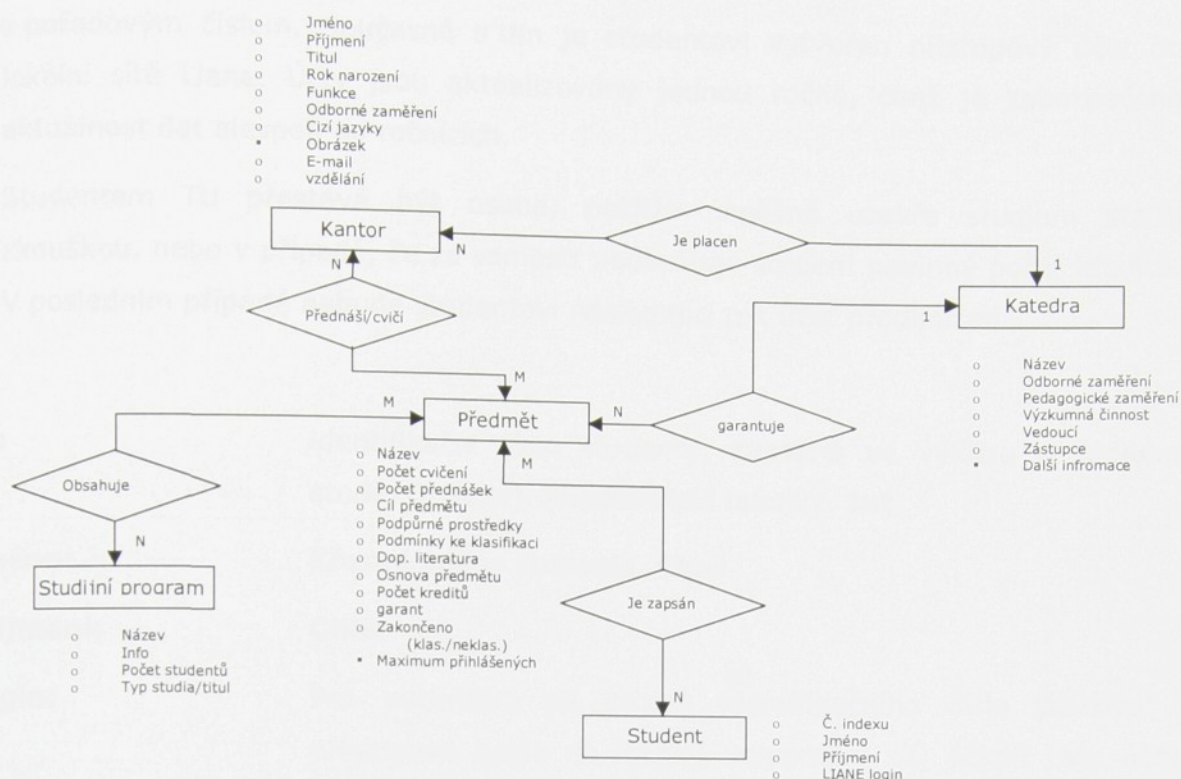
5.3 Analýza

Studium na hospodářské fakultě přechází na novou, volnější formu – tzv. kreditový systém. Studenti již nejsou vázáni pevným studijním řádem, jako tomu bylo v minulosti, ale mají možnost si studium rozdělit podle vlastních aktuálních dispozic. Důraz je v souvislosti kladen i na tzv. volitelné předměty. Nový informační systém by měl řešit problém registrace studentů na volitelné předměty. Zároveň by jim měl dát ucelený obrázek o budoucím studiu na fakultě. Kromě klasického studijního programu, by měl mít student k dispozici informace o vyučujících, doporučené literatuře, přibližné osnově předmětů apod. Do budoucna by studenti měli mít možnost on-line evaluace cvičících a přednášejících.

Jelikož se ve studijním plánu kryjí některé předměty pro více studijních programů – Jeden předmět je vyučován pro posluchače různých studijních programů, měl by systém počítat i s touto eventualitou a zamezit duplicitním datům.

5.3.1 Konceptuální model

Výsledkem této fáze je model reálného systému, tzv. konceptuální schéma databáze. Pro vytvoření konceptuálního schématu se hodí tzv. **Entity-Relationship-Attribute** model. V následujícím obrázku je definován základní model systému pomocí pěti základních entit, jejich atributů a vzájemných vztahů.



obrázek 5 - ERA diagram

Jak z obrázku vyplývá, převažují v našem konkrétním případě vztahy M:N. Příkladem budiž vztah entit student a předmět. Jeden student může být zapsán na několik předmětů, stejně tak, jako na jeden předmět bývá zapsáno více studentů.

V grafu chybí například vztah předmětů navzájem (pro definování vztahu návaznosti jednotlivých předmětů). V grafu by sice mohla být entita předmět zobrazena dvakrát, popisující návaznost předmětu na jiný, ale pro přehlednost jsem tak neučinil.

5.3.2 Logická struktura datové základny

Tato fáze spočívá v implementaci konceptuálního modelu.

Z předchozího ERA diagramu jsem vycházel při definování vlastní struktury databáze.

5.3.2.1 Student

Pod entitou student rozumějme platného studenta hospodářské fakulty. Platnému studentovi je při zápisu do prvního ročníku vydán výkaz o studiu na TU v Liberci

s pořadovým číslem. Současně s tím je studentovi vytvořen přístupový účet do lokální sítě Liane. Účty jsou aktualizovány jednou ročně, čímž je zabezpečena aktuálnost dat alespoň po ročnících.

Studentem TU přestává být osoba, pakliže úspěšně uzavře studium státní zkouškou, nebo v případě, že za semestr nedosáhne student povinný počet kreditů. V posledním případě nebude studentovi následující rok účet prodloužen.

Id:	identifikační číslo studenta, zapsané ve výkazu o řádném studiu (indexu). Primární klíč tabulky. Int
Jmeno:	Křestní jméno studenta. Char
Prijmeni:	Char
Login:	Pro autorizaci se kromě identifikačního čísla používá i přístupové heslo na lokální univerzitní síť. Ačkoli jsou účty vytvářeny podle šablony jméno_příjmení , existují případy, kdy login není totožný s Jméno . „_“ . Příjmení. Uvedme třeba případ, kdy se studentka během studia vdala, nebo případy totožných jmen i příjmení u více studentů. Z tohoto důvodu musel být vytvořen další atribut entity student. Char

5.3.2.2 Kantor

Pod typem entity kantor chápeme zaměstnance hospodářské fakulty, kteří buď přednáší, nebo mají na starosti vedení seminářů (nebo oboje). Existence výskytu entity je spojena s pracovní smlouvou. Vzhledem ke způsobu aktualizace dat týkajících se studentů (pro účely registrace volitelných předmětů), jsem se nerozhodl použít pro studenty a kantory společnou tabulku s rozlišujícím atributem. V současné době je varianta oddělených tabulek přijatelnější.

Id:	Pořadové číslo kantora. Tento atribut „navíc“ řeší problémy redundance a nutnost kombinovat klíče. Primární klíč tabulky. Int
Jmeno:	Char

Prijmeni:	Char
Titul:	Char
Funkce:	Char
	Ačkoli se často vyskytují hlavní tři kategorie, nejedná se o úplný výčet:
	- Odborný asistent - Profesor - Vedoucí katedry
Rodne cislo:	Char
Odb_zam:	Odborné zaměření. Text
Jazyk:	= jazykové znalosti, výčet jazyků. Char
Image:	= odkaz na fotografii. Char
Pub_cin:	= publikační činnost. Text
Vzdelani:	Dosažené vzdělání. Char
Email:	Adresa elektronické pošty. Char
Kat_id:	Atribut zajišťující 1:N vztah k entitě Katedra. Int
Admin:	Boolean atribut definující, jestli má osoba přístup k aktualizaci katedrových dat.
Nastup:	Datum, určující, jak dlouho již pedagog na HF působí. Date

5.3.3 Katedra

Katedra je nejnižší organizační jednotkou Technické univerzity. Každý pedagog vyučující na HF TU v Liberci spadá pod jedinou katedru.

Id:	Identifikační číslo katedry. Primární klíč. Int
Nazev:	Plné jméno katedry. Char
Kod:	Zkratka katedry. Většinou třípísmenná. Char
Odb_zam:	Odborné zaměření katedry. Text

Ped_zam:	Pedagogické zaměření katedry. Text
Vyzkum:	Výzkumná činnost katedry. Dokončené i probíhající výzkumné projekty. Text
Ostat_info:	Další nezařazené informace o katedře. Text
URL:	Odkaz na domácí WWW stránku katedry. Char
Vedouci:	Id katora/vedoucího katedry. Int
Zastupce:	Id kantora/zástupce vedoucího katedry. Int

5.3.4 Studijní program

Entita „studijní program“ determinuje náplň studia, studijní program studenta během svého studia na Technické univerzitě, včetně délky studia a typem ukončení vysokoškolského studia (inženýrské, bakalářské, magisterské ...).

Id:	Id studijního programu. Int
Nazev:	Označení studijního programu. Char
Semestru:	Počet semestrů. Int
Titul:	Titul příslušící absolventovi studijního programu. Char
Vedouci:	Id kantora/zástupce vedoucího katedry. Int

5.3.5 Předmět

Id:	Id předmětu. Int
Název:	Název předmětu podle studijního plánu. Char
Kat_id:	Katedra garantující daný předmět. Int
Prednasky:	Počet přednášek za týden. Int

Cvícení:	Počet cvičení za týden. Int
Cíl:	Cíl předmětu. Stručná charakteristika, proč byl předmět do studijního programu zařazen. Char
Pp:	Podpůrné prostředky, pomůcky apod. Char
Podm:	Podmínky udělení kreditu. Char
Liter:	Doporučená literatura. Char
Osnova:	Stručná osnova předmětu. Char
Zakonceno:	Typ zakončení předmětu – klas./neklas. Bool
Maxlidi:	Maximální počet posluchačů. V případě, že se jedná o volitelný předmět, má tento atribut nenulovou hodnotu. Int
Kredit:	Počet kreditů. Int
Garant:	Osoba odpovědná za náplň vyučovaného předmětu. Vztah na tabulku kantor. Int

Součástí databáze jsou též spojovací tabulky zabezpečující vztah M:N. Tento vztah je definován pomocí kombinací primárních klíčů tabulek entit majících vztah M:N.

V této fázi je na místě rozhodnout o další možné dekompozici jednotlivých tabulek. Je vhodné nechat položku „funkce“ v tabulce kantor, nebo vytvořit samostatnou tabulku funkcí?

Osobně se domnívám, že je celý systém natolik flexibilní, že i v případě nutnosti pozdějších úprav není dekompozice existujících tabulek nemyslitelná a současný návrh považuji v tuto chvíli za akceptovatelný.

6. POPIS SYSTÉMU

Tato kapitola podrobně popisuje postup administrace systému. Je základem dokumentace pro budoucí administrátory katedrových dat.

Nový systém umožňuje osobě pověřené vedoucím katedry aktualizovat informace bez nutnosti kontaktovat správce serveru. Katedry tak získávají možnost reflektovat okamžitě změny týkající se zaměstnanců katedry, vyučovaných předmětů apod.

K vlastním úpravám v datech týkajících se katedry je pověřen buď vedoucí dané katedry, nebo osoba vedoucím katedry pověřená. Z důvodů odpovědnostních není možné definovat více osob oprávněných k zápisu.

Klíčovou obsahovou částí jsou informace o vyučovaných předmětech. Data o předmětech jsou ale těsně svázána s daty o katedře, studijním programu a vyučujících. K tomu, aby bylo možné vkládat nová data, je nutné doplnit nejprve data týkající se katedry a personálu.

Klíčovým momentem je přenesení břemena aktualizace a odpovědnosti za ní z jedné osoby na jednotlivé katedry, konkrétně na vedoucího katedry nebo osobu pověřenou vedoucím dané katedry.

Tato pověřená osoba tedy vystupuje jako administrátor dat týkajících se katedry. Jelikož je systém uživatelsky "přítulný", není na správce kladen nijak extrémní nárok na seznámení se se systémem. V této kapitole se pokusím popsat a vysvětlit všechny funkce a možnosti které má administrátor katedrových dat k dispozici.

6.1 Autorizace

K tomu aby byla zaručena alespoň standardní úroveň bezpečnosti, musí se administrátor obsahu do systému přihlásit. Při volbě položky "vedoucí kateder" z hlavního menu je uživatel dotázán na přístupové jméno a heslo. K autorizaci je použito již existujících účtů do lokální sítě Liane. To s sebou sice přináší jisté bezpečnostní problémy (blíže viz kapitola 4.5 - Bezpečnost), nicméně neklade další nároky na administrátora katedrových informací, který by si musel zapamatovat další heslo a administrátora celého systému, který by účty musel zřizovat.

V případě úspěšné autorizace je uživateli nabídnuta stránka kde si může vybrat z následujících variant:

- přidat zaměstnance,
- přidat předmět,
- odebrat zaměstnance,
- odebrat předmět,
- opravit zaměstnance,
- opravit předmět,
- opravit informace o katedře,
- přidat/odstranit/opravit nástěnku.

Po zvolení patřičné položky potvrdí administrátor katedrových dat svůj výběr tlačítkem "pokračovat".

6.2 Přidat zaměstnance

Před tím, než bude administrátor přidávat data týkající vyučovaného předmětu je nutné, aby databáze obsahovala data o všech vyučujících (tj. jak přednášejících, tak cvičících).

Po vybrání položky "přidat zaměstnance katedry" z hlavního menu, se v hlavním okně prohlížeče objeví formulář. Všechny položky označené hvězdičkou jsou údaje povinné. Pakliže administrátor zapomene některou takovouto položku vyplnit, bude na tuto skutečnost upozorněn a bude ji muset zpětně doplnit. Ostatní položky nejsou povinné, nicméně je žádoucí, aby i tato data byla v databázi obsažena.

Všechna políčka mluví sama za sebe, takže nemyslím, že by bylo nutné se u nich zastavovat. Jedinou částí zasluhující pozornost je položka "obrázek". Ačkoli situace, kdy administrátor katedrových dat bude sám definovat odkaz na obrázek zaměstnance, není příliš standardní, může v budoucnu nastat například, když bude mít katedra k dispozici lepší, novější fotografie.

Pro definici odkazu na vlastní obrázek je nutno použít absolutní URL identifikátor. (pakliže nezačíná na řetězec „http://“ nebo „ftp://“, bude použita implicitní cesta a obrázek nebude korektně zobrazován)

Pakliže není fotografie v momentě zařazení zaměstnance k dispozici, stačí nechat v políčku "obrázek" implicitní nastavení "none.gif".

6.3 Přidat předmět

Poměrně důležitou sekci formuláře je výběr studijního programu, kterého se předmět týká. Díky novému kreditovému systému má student možnost vybrat si, v jakém semestru bude předmět navštěvovat. Navíc je časté, že je jeden a tentýž předmět vyučován pro více studijních programů. Tato část formuláře je tedy koncipována následovně.

Pomocí zatržitek (select box) administrátor vybere jednak studijní programy, pro které se předmět vyučuje (alespoň jeden) a k příslušným studijním programům vybere stejným způsobem i semestr, který je podle studijního plánu doporučen aby jej student daný semestr studoval. Pakliže bude vybrán pouze určitý semestr bez vybrání příslušného studijního programu, nebude na příslušné semestry brán zřetel.

6.4 Odebrat zaměstnance

V případě, že některý vyučující opustí katedru, je možné ho odstranit. Když zaměstnanec ale pouze přechází na jinou katedru, je elegantnějším řešením převod zaměstnance hlavním administrátorem systému. Pakliže chce administrátor katedrových dat ulehčit práci sobě i svému kolegovi z druhé katedry, stačí zaslat elektronickou poštou informace o změně zařazení zaměstnance a ten pak provede jednoduchou změnu této skutečnosti. Postup kdy jsou data vymazána a posléze znovu zaváděna jinde je zbytečně komplikovaný.

Když ovšem zaměstnanec opouští Hospodářskou fakultu, pro jeho odstranění z databáze slouží právě tato volba. Administrátor dostane výpis všech zaměstnanců katedry, ze kterých vybere příslušnou položku a po potvrzení výběru je tento záznam vymazán.

6.5 Odebrat předmět

Když daný předmět již není vyučován a neexistuje „následník“, (v tom případě by bylo snazší pouze změnit data, ne je mazat a vytvářet...) je možné tento předmět z databáze vyřadit.

Kromě vlastního předmětu jsou odstraněny záznamy týkající se vazeb na přednášející, cvičící, jiné předměty, přihlášené studenty, atd.

6.6 Opravit zaměstnance

Pakliže dojde ke změně skutečností týkajících se zaměstnance, je možné tyto změny okamžitě zavést do systému. V podstatě je postup totožný s tím, uvedeným v kapitole 6.2 - Přidat zaměstnance, s tím, že položky formuláře již jsou předvyplněny existujícími daty. Před vlastní úpravou dá systém vybrat, kterého zaměstnance chcete upravovat.

6.7 Opravit předmět

Při opravách předmětů je postup obdobný jako u oprav zaměstnanců. Pro větší přehled při výběru přednášejících, cvičících a návazných předmětů je u názvu položky v závorce počet vybraných položek. Administrátor má tak přehled, že už jsou vybrána určitá data a může tak zamezit nechtěnému zrušení tohoto výběru.

7. PODNIKOVÝ IS

7.1 IS jako konkurenční výhoda

Stejně jako byla před nedávnem pozornost managementu směřována na značku, kvalitu, pověst, se dnes za konkurenční výhodu považuje způsob, jak organizace pracuje s informací. V dnešní době již většina výrobců pracuje se stejným materiálem, na stejných obráběcích strojích, s podobnými výrobními procesy. Ne že by všechny zmíněné faktory nebyly důležité – naopak, jsou podmínkou úspěchu. Stejně tak se i informační systém z konkurenční výhody stává strategickou nutností.

Existují dokonce obory na informacích postavené, jako bankovníctví a pojišťovnictví, kde dobře fungující IS je klíčovým faktorem úspěchu.

Při posuzování hodnoty investic do IS narážíme na problém těžké kvantifikace přínosů informačního systému. Informační systémy s sebou přináší spíše kvalitativní změny, které ovšem v klasickém pojetí investičního rozhodování nejsou exaktně definovatelné. Pakliže na IS hledíme pouze jako na prostředek ušetření nákladů spojené s automatizací existujících procesů, uniká nám hlavní výhoda aplikací IS. IS dnes otevírají možnost oslovit nové zákazníky, proniknout do nových segmentů trhu, zlepšení komunikace se zákazníky, vznik nových služeb, atd. V této poloze fungují IS jako strategický nástroj a dávají organizaci konkurenční výhodu.

Společnosti, které využívají IS/IT, aby inovovaly způsob, jakým vyvíjejí svou činnost, a za účelem růstu, dosahují vyšší návratnosti svých investic do IS/IT než organizace, které usilují pouze o zlepšení efektivity provozu.

Posuzovat hodnotu IS/IT pomocí nástrojů finanční analýzy podhodnocuje přínos těchto systémů. Hovoříme-li o informačním věku, znamená to, že minimálně 50% národního produktu vyspělých zemí je tvořeno tzv. informační prací. [5] Bez nástrojů pro podporu rozhodování není možné tuto práci vykonávat dostatečně efektivně.

7.2 Kontinuita hodnoty IS/IT

Metodika s názvem „Kontinuita hodnoty“ navrhuje dvě překrývající se kategorie hodnoty IT: základní hodnotu a inovační hodnotu se dvěma podtypy hodnot v každé

z těchto kategorií. Základní hodnota je tvořena efektivitou a účinností a inovační hodnota zahrnuje expanzi trhu a vytvoření výhody.

7.2.1 Základní hodnota

- **Efektivita** – Pokud je prvořadým cílem, organizace usiluje o to, aby procesy dosahovaly vyšších rychlostí při nižší četnosti chyb a nižší úrovni nákladů. Tento druh prospěchu lze získat pomocí automatizace základních úloh, k nimž patří výplata mezd nebo informování zákazníků o změně ceny pomocí faxu/elektronické pošty.
- **Výkonnost** – Zde se informační systémy používají především pro zvýšení provozní výkonnosti. Společnost může např. použít databázi nejlepších praktik používaných v rámci řízení vztahů se zákazníky, za účelem podpory transferu znalostí mezi jednotlivými profesionálními disciplínami či podnikatelskými jednotkami.

7.2.2 Inovační hodnota

- **Expanze trhu** – V tomto případě je cílem investovat do rozvojových projektů systémů, které slibují lepší růst podílu společnosti na existujících trzích. Například společnost dodávající spotřební zboží a poskytující služby může vybudovat moderní databázi a použít techniky pro modelování zákaznického segmentu, aby předpověděla, jaké výrobky bude zákazník kupovat.
- **Vytvoření výhody** – Zde organizace používá IS/IT jako platformu pro inovaci. Někteří výrobci automobilů používají systémy založené na službě WWW pro objednávkový systém pro konečného zákazníka. Ten si může vybrat vozidlo přesně podle svých představ bez jakýchkoliv časových prodlev.

7.3 Strategie použití IS/IT

- Exkluzivní výhoda – organizace pracuje s vlastní technologií, která jí poskytuje konkurenční výhodu a zároveň tuto technologii navenek chrání (např. pomocí patentů, náročnosti na znalosti apod.).
- Jeden krok napřed – náročná strategie při které se organizace snaží o kontinuální vývoj a použití nových technologií. Tato neustálá inovace dává organizaci náskok před konkurencí. Systémové změny vedoucí ke kontinuálnímu snižování objednacích lhůt mohou dávat organizaci výhodu nad konkurencí.

- Diskontinuita – Vysoká inovační hodnota nové technologie umožní organizaci skok, který není opakovatelný. Např. když Citibank přišla se sítí bankomatů, znamenala tato inovace téměř ztrojnásobení tržního podílu v USA. A i když další bankovní domy následovaly, k návratu zákazníků to nevedlo.
- Implementace – organizaci se podaří běžně dostupnou technologii nadprůměrně dobře implementovat.

7.4 Náklady IS

Mluvíme-li o nákladech spojených s IS, musíme si tyto náklady rozdělit do kategorie nákladů skrytých a patrných.

tabulka 3 - porovnání nákladů vlastního projektu a zakoupeného "balíku"

Projekt realizován vlastními silami			Zakoupený systém		
náklady	předpokládané	skryté	náklady	předpokládané	skryté
Plánování a návrh	40%		Plánování a návrh		40%
Programování	10%		Analýza/ozkoušení	25%	
Testování	10%	20%	Testování	5%	10%
Instalace	10%	10%	Instalace		20%
Celkem	70%	30%	Celkem	30%	70%

Náklady na IS stejně jako každý jiný náklad je třeba řídit, tedy minimalizovat nikoli na úkor výnosů. Zkušenosti s plánováním nákladů spojených s IS jsou spíše negativní, právě z důvodů existence příliš mnoha skrytých nákladů. [6] Všeobecně se dá říct, že systémy tvořené vlastními silami jsou méně nákladné, než zakoupené systémy „na klíč“, neboť nesou daleko menší procento skrytých nákladů. Nesprávně odhadnutá míra nákladů může vést:

- ke zvýšené míře nedůvěry k IS,
- jsou vybírány projekty, které by vybírány být neměly,
- kvůli promrhaným zdrojům není investováno do vhodných projektů.

Rozdíl mezi investičním rozhodováním do IS oproti ostatním projektům je zásadní – aby investice mohla být posouzena, musí předem existovat podrobná znalost projektu. Tato podmínka často není a nemůže být splněna. Rozhodnutí o investicích

do IS jsou často posuzována na základě informací, přirovnatelných ke stejné úrovni detailu, jako při rozhodování o stavbě nové budovy na základě shlédnutí pozemku stavby.

Nákladová prognóza navíc musí pracovat s určitým množstvím historických dat, a ty v souvislosti s IS často vůbec neexistují, protože se jedná o první projekty svého charakteru.

Zjednodušeně si můžeme rozdělit druhy nákladů spojené s IS do několika následujících kategorií:

1. **Náklady na hardware:** veškeré zařízení nutné pro realizaci projektu IS. Většinou se předpokládané náklady na hardware příliš nemění.
2. **Náklady na software:** Nejlevnější alternativou jsou již hotové softwarové produkty všeobecného zaměření, použitelné obecně. Náklady spojené s vlastním vývojem nejsou skoro nikdy přesněji odhadnutelné.
3. **Náklady spojené s instalací:** Instalační náklady se různí systém od systému, stejně tak i náklady s tím spojené. Často se do nákladů na instalaci započítávají náklady spojené s převodem dat z existujícího systému do nového (nejmarkantnější je tato položka při přechodu od manuálního zpracování).
4. **Náklady na zabezpečení prostředí:** Do této kategorie spadají náklady na klimatizaci, nábytek, kabeláž apod.
5. **Provozní náklady:** telefonní poplatky, elektřina, poplatky za poskytování připojení k internetu, poplatky za využívání online služeb jako jsou všemožné databáze apod.
6. **Náklady na údržbu:** Ať už se jedná o údržbu plánovanou, či podle potřeby, hardwarovou, či softwarovou...
7. **Bezpečnostní náklady:** Náklady na zabezpečení systému proti rizikům ohrožujícím funkčnost systému.
8. **Náklady na údržbu datové základny.**
9. **Náklady na školení:** Tato část nákladů je v drtivé většině případů podceňována a často se v projektech počítá se získáním zkušeností se systémem jako s nedokumentovanou samozřejmostí.

10. **Širší organizační náklady:** Sem jsou zařazovány náklady spojené s nekompatibilitou, náklady spojené s přechodem na nový systém, změnami ve stávající organizaci podniku.

Investice do informačních systémů nejsou promptní investicí. Jakmile organizace začne s investicí do IS, nejde o jednorázovou položku jako při investici do nového stroje, nebo budovy. Investice do IS jsou investicemi průběžnými a milionový roční projekt je ve skutečnosti několikamilionovým dlouhodobým projektem zabezpečujícím funkčnost systému po celou dobu jeho používání. Bez dodatečných investic by IS nebyl funkční a investice se stala naprosto zbytečnou. Ve většině případů se s těmito dodatečnými náklady ale vůbec nepočítá. Vize investice do IS jako konečném investičním projektu je v zásadě chybná a je nutné počítat s rostoucími „udržovacími“ náklady.

Rozlišme dva základní pohledy na náklady spojené s IS:

Prostředky vynaložené na IS jako výdaj:

1. Minimální výdaje.
2. Operační pohled na výhody IS.
3. Opravdu to musíme utrácet?
4. Konečný, nekontinuální výdaj.
5. Nákladová analýza.
6. Vlastní financování pomocí úspory nákladů v jiných oblastech.
7. Zaměření na stránku nákladovou.

Prostředky vynaložené na IS jako investice:

1. Velké finanční výdaje.
2. Strategické výhody.
3. Do toho musíme investovat!
4. Kontinuální investice.
5. Posouzení investice.
6. Kapitálové plánování.
7. Zaměření na stránku výnosů/výhod.

7.5 Hodnocení investic do IS

V této části bych se rád pozastavil nad několika metodami posuzování investic do IS. Proč je vlastně toto posouzení tak důležité? Organizace sledují posuzováním/hodnocením investic do IS/IT následující:

- Odůvodnění investic.
- Možnost provést rozhodnutí mezi konkurenčními projekty.
- Posuzování investic funguje jako kontrolní mechanismus výdajů, výnosů vývoje a implementace jednotlivých projektů.
- Získání dostatečného množství znalostí pro další posouzení investic v budoucnu.

7.5.1 Techniky pro kapitálové posouzení investic

Podle průzkumů v Británii⁷ a USA v letech 90-93 se stále nejčastěji používalo jednoduchých metod investičního rozhodování, jako je metoda návratnosti (doba úhrady). Složitější metody, jako metoda čisté současné hodnoty (NPV – net present value), metoda vnitřního výnosového procenta (Internal rate of return – IRR), za tu dobu nabývaly popularity a v dnešní době patří mezi běžně používané.

Payback⁸ – Doba úhrady se definuje jako doba potřebná pro úhradu celkových investičních nákladů projektu jeho čistými výnosy, kde jako čisté výnosy projektu se chápe součet zisku po zdanění a odpisů (v případě, že se projekt financuje i z cizích finančních zdrojů, např. bankovním úvěrem, tvoří součást čistých výnosů i úroky z tohoto úvěru).

Stanovená doba úhrady projektu se pak porovnává s její určitou normovanou (mezí) hodnotou, zvolenou firmou (obvykle na základě minulé zkušenosti a ostatních investičních příležitostí). Pokud je doba úhrady projektu nižší než doba mezí, jedná se o projekt vhodný k realizaci. Velikost rozdílu obou

⁷ Ballantine J., Stray S.: Journal of Information technology 1998/13, Financial appraisal and the IS/IT investment decision making process, str 3.

⁸ Fotr, J.: Podnikatelský a investiční rozhodování. 1. Vyd. Praha 1995, str. 89-101

hodnot určuje míru výhodnosti takového projektu. Při porovnávání více podnikatelských projektů je nevhodnější projekt s nejkratší dobou úhrady.

Hlavní předností této metody je její srozumitelnost a jednoduchost, na druhé straně však naprosto nerespektuje faktor času (viz následující metody), ignoruje strukturu výnosů po době úhrady a navíc až příliš zdůrazňuje rychlost návratnosti projektu.

Metody založené na diskontování:

NPV – Net Present Value, neboli čistá současná hodnota se definuje jako součet diskontovaných čistých toků hotovosti během doby života projektu. Vzorek pro výpočet této metody můžeme zapsat jako:

$$\check{C}SH = \sum_{i=1}^n \frac{\check{C}TH_i}{(1+r)^i}$$

kde:

$\check{C}SH$ čistá současná hodnota projektu
 $\check{C}TH$ čistý tok hotovosti v i-tém roce života projektu
 r diskontní sazba

Jako diskontní sazbu stanovíme minimální výnosovou míru, se kterou investor minimálně počítá. Pro přepočtení této nominální hodnoty na hodnotu reálnou, budeme potřebovat znát ještě průměrné tempo inflace.

$$r = \left(\frac{1+n}{1+m} - 1 \right)$$

kde:

r reálná diskontní sazba
 n nominální diskontní sazba
 m míra inflace

IRR – Vnitřní výnosové procento se chápe jako výnosnost (rentabilita), kterou poskytuje podnikatelský projekt během doby svého života. Číselně je vnitřní výnosové procento rovno takové diskontní sazbě, při které je čistá současná hodnota projektu rovna nule. Jelikož se při výpočtu jedná o rovnici n-tého stupně, málokdy se počítá IRR ručně. S nástupem tabulkových kalkulátorů však i tato metoda nabývá popularity.

Uplatnění vnitřního výnosového procenta i čisté současné hodnoty vede k týmž závěrům.

Problémy s použitím výše uvedených metod při posuzování investic do IS/IT však nejsou nijak dokumentovány.

tabulka 4 – Použití různých metod posouzení investic v Británii⁹

metoda/autor (výzkum v r.)	Sanster 1989 (%)	McIntyre / Coulthurst 1984 (%)	Mills / Herbert 1984 (%)	Pike 1980/81 (%)	Pike 1975 (%)
Doba úhrady	70	82	78	79	71
IRR	58	28	68	54	42
NPV	48	36	51	38	32
ARR ¹⁰	31	33	44	51	51

Důvody proč nepoužívat výše uvedené metody při posuzování investičních projektů do IS/IT nenajdeme ve finanční nebo účetní literatuře, ale samozřejmě v literatuře zabývající se informačními systémy. Vychází to z přesvědčení že investice do IS/IT je kvalitativně odlišná od kterékoli jiné investice. V případě investic do IS je velmi složité předpovídat budoucí výnosy, na kterých jsou klasické metody založené stejně jako na vyčíslení investičních nákladů. S investicemi do IS dále souvisí časté neočekávané výhody, které ve většině případů vůbec nejsou kvantifikovatelné.

Většina finančních metod je primárně orientována na zisk. To je, samo o sobě, v rozporu s intencí investic do IS. Projekty jako „zlepšení podpory zákazníka“, nebo „zlepšit informovanost o trhu“ nejsou z krátkodobého hlediska finančně zhodnotitelné.

Metody jako NPV, IRR by byly použitelné pouze v případě znalosti budoucích finančních toků, což v případě IS/IT jsou jen velmi hrubé odhady.

Na druhé straně by měla existovat nějaká obecná metoda finančního posouzení jakéhokoli investičního projektu. Nejde přeci o jediný rozhodovací faktor. Musíme si uvědomit, že všechny výše uvedené metody nejsou striktním vodítkem, která řeší

⁹ Ballantine J., Stray S.: Journal of Information technology 1998/13, Financial appraisal and the IS/IT investment decision making process, str 3.

¹⁰ Průměrná doba návratnosti

otázku zda investovat, či nikoli. Je pouze jedním z podpůrných hledisek v tomto rozhodovacím procesu.

7.6 Koupit nebo vyrobit

Problémem, který řešila, nebo spíše neřešila spousta českých podniků, postavená před problém nefungujícího, zastaralého informačního systému – bylo posouzení, zda zakoupit hotový balík od většinou zahraničního dodavatele, nebo se pustit do vývoje vlastního IS. Tedy z pohledu životního cyklu informačního systému, skočit rovnou do implementační fáze, v případě zakoupeného systému, nebo projít všemi fázemi v případě vlastního řešení.

7.7 Nákup hotového „balíku“

V tuto chvíli úplně pominu funkční stránku věci, což samo o sobě je obrovská chyba. K čemu je levnější řešení, které nesplňuje požadavky na systém kladené? V případě zakoupení hotové IS se organizace řídí při koupi zejména informacemi z tisku, zkušenostmi konkurence (pakliže jsou v určité podobě k dispozici a chcete jim věřit), a informacemi přímo od distributora onoho systému. V mnoha případech jsou informace nutné k posouzení použitelnosti daného systému pro účely organizace nedostačující a rozhodnuto je na základě nepodstatných¹¹ atributů nabízeného systému.

Navíc pro „hotové balíky“ na první pohled mluví nižší náklady, respektive není u vlastního vývoje předem známa konkrétní výše nákladů a vše je jen hrubě odhadováno, zatímco u nakupovaného řešení má management k dispozici konkrétní číslo. U nakupovaného řešení však organizace musí počítat s nezanedbatelnými náklady skrytými, které jsou ve většině případů spojené s implementací IS do konkrétní organizace a změnami existujících procesů v organizaci. Tyto změny jsou zřídkakdy toho charakteru, aby došlo k zefektivnění těchto procesů. Většinou se jedná o změny, které jsou nutné k tomu, aby byl model systému aplikovatelný na danou organizaci. Tedy zcela obrácený postup než v ideálním případě, kdy se hledá

¹¹ Za nezásadní informaci považuji i čísla o množství implementací v různých podnicích, neboť to naprosto nevyjadřuje jak úspěšně tam systém funguje. Navíc skvěle fungující systém pro plánování zásob může skvěle fungovat v chemičce, nikoli pak třeba ve firmě zabývající se zneškodněním odpadů.

optimální proces a upravuje se podle něj kýžený informační systém. Samozřejmě, že s implementací se jen zřídka dokáže organizace vypořádat sama, takže nastupují zkušenější, ale také velmi drahé, konzultační firmy.

Samozřejmě, že existuje možnost, kdy zakoupený systém „sedne“ přesně na potřeby organizace. Bohužel ve fázi posouzení koupě toho či onoho IS není k dispozici dostatek informací pro učinění správného rozhodnutí. Zatím jsem se nesetkal s komerčním systémem, který by organizace mohla na zkoušku otestovat a v případě negativních zkušeností bez sankcí odmítnout.

Ceny komerčních produktů jsou skutečně různé – od několika tisíc za jednodušší systémy na bázi finančního účetnictví až po milionové sumy za systémy zahrnující finanční účetnictví, controlling, lidské zdroje, logistiku, plánování výroby, prodej a distribuci, údržbu, řízení kvality, řízení dat týkajících se produktu apod. Nabídka informačních systémů na bázi podnikového intranetu – tedy vesměs IS založené na uživatelském rozhraní WWW a SQL databázi dosahuje doslova boomu. Se svými nabídkami přichází nejenom giganti typu IBM, ale s levnějšími alternativami přichází spousta i domácích firem. Technických prostředků je nespočet. Stejně jako se v nedávné minulosti vesměs každá softwarová firma v tehdejší Československu zabývala vývojem nějakého účetního systému, se dnes firmy vrhají na poskytování „komplexního“ informačního systému pro organizace. Málokterá taková firma ale podrobně rozumí procesům v dané organizaci, a ačkoli modelově systém funguje, v mnoha organizacích je naprosto neimplementovatelný. Jestli zakoupený systém bude bezproblémově fungovat záleží na tom, jak se konkrétní organizace liší od modelové organizace se kterou vývojová firma při tvorbě systému pracovala.

7.8 Vlastní vývoj

Když se organizace rozhodne pro vývoj vlastního systému „na míru“, počítá předem s patrnými náklady jako mzda vlastním analytikům, programátorům, externím konzultantům apod. Ve fázi proveditelnosti je předložen předběžný nástin nákladů na hardware, software. Jelikož cílem celého projektu je zkvalitnění určitých procesů, během implementace vyjdou najevo další skryté náklady spojené se změnou stávajících procesů.

Pakliže si podnik jako technickou základnu vybere stejné prostředky, které byly použity při návrhu informačního systému Hospodářské fakulty, ušetří

nezanedbatelné prostředky; tabulka 5 dokumentuje řešení založené na technologiích společnosti Microsoft.

Řešení založené na platformě Linux, kromě naprosto otevřených licenčních podmínkách přijde na 0-\$250. (\$50 za komerční distribuci OS Linux od společnosti RedHat a \$200 za licenci MySQL pro komerční využití produktu).

Samozřejmě, že náklady na technické prostředky nejsou nejpodstatnější položkou rozpočtu na IS podniku, ale v případě nutnosti většího počtu licencí je tato částka již markantní. Ušetřené prostředky je pak vhodnější vložit do důkladnější analýzy.

tabulka 5 – cena technického řešení na platformě Windows NT

Operační systém	Microsoft Windows NT Sever 4 verze pro 5 uživatelů	\$809
WWW server	IIS	V ceně
Server-side scriping	Active Server Pages	Zdarma
DBMS	MS SQL server 7 – pro 5 klientů (dodatečné licence po 5 za \$739	\$1 399

8. ZHODNOCENÍ

Cílem mé diplomové práce bylo vytvoření systému komunikace Hospodářské fakulty se studenty.

V době psaní těchto řádků se z pohledu životního cyklu systému nachází ve fázi implementace. V laboratorních podmínkách byl již vyzkoušen a byly opraveny chyby v uživatelském rozhraní a byly provedeny některé další změny.

Studenti budou mít v nejbližší době k dispozici podrobné informace o studijním plánu, jednotlivých předmětech, vyučujících. Mají možnost se přihlásit na volitelné předměty. K dispozici je také seznam absolventů Hospodářské fakulty. Studenti mohou diskutovat o studijních problémech pomocí systému nástěnek. Existují speciální nástěnky, kde mohou podniky nabízet volná pracovní místa apod. Hlavní přínos celého systému ale tkví ve způsobu aktualizace, který je decentralizovaný a umožňuje jednotlivým katedrám kontrolovat obsah části týkající se dané katedry.

Je nesmírně důležité, aby si na nový systém jednotlivé katedry zvykly. Celý systém je založen na požadavku aktuálnosti dat. Nyní má každá katedra povinnost starat se o data týkající se konkrétně katedry. Celý proces přijde nazmar, pakliže se nenajde způsob jak donutit pověřené osoby tuto aktualizaci skutečně provádět.

V současné implementační fázi musí být kladen důraz na školení a seznámení všech vedoucích kateder s novým způsobem aktualizace IS.

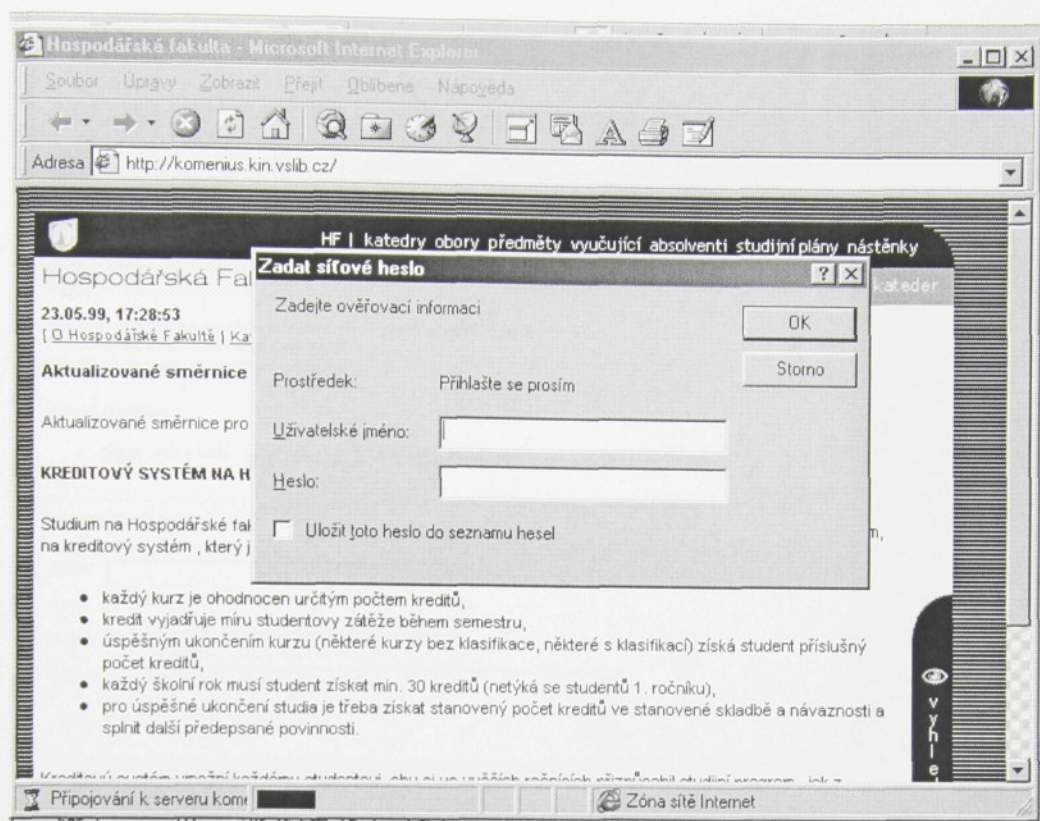
Celý systém je veřejně dostupný na adrese <http://komenius.kin.vslib.cz>.

POUŽITÁ LITERATURA

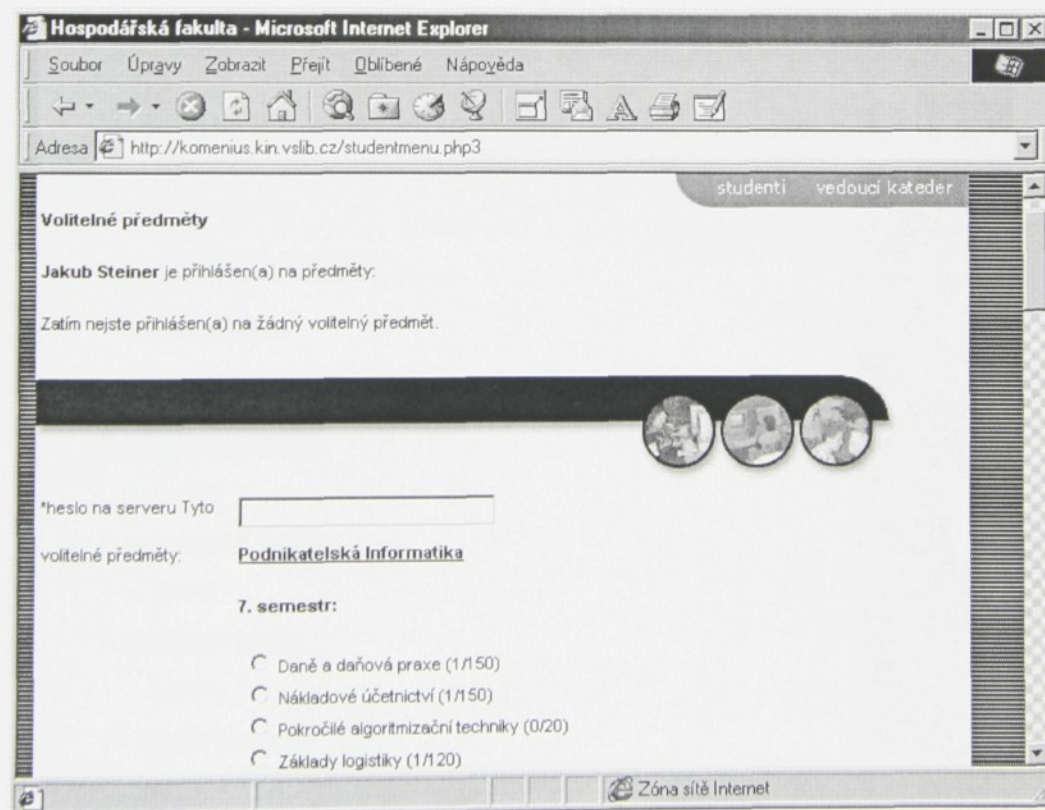
- [1] CERT: Security for a public WebSite. Carnegie Mellon University 1997
- [2] SATRAPA, P., RANDUS, J. A.: Linux – Internet Server. 2. vyd. Neokortex 1996, ISBN 80-902230-0-1
- [3] KENDALL, P.A.: Introduction to Systems Analysis and Design: a Structured Approach. 2. vyd. WCB 1992 ISBN 0-697-07748-9
- [4] STRAKA, M.: Vývoj databázových aplikací. 1.vyd. Praha 1992, Grada, ISBN 80-85424-43-6
- [5] GATES, W.: Byznys rychlostí myšlenky. 1. Vyd. Ringier ČR a.s. 1999, ISBN 80-85943-97-2
- [6] ROBSON, W.: Strategic management and Information Systems, an integrated approach. 1. vyd. Pitman Publishing 1994, ISBN 0-273-60042-7
- [7] SATRAPA, P.: World Wide Web. 2. vyd. Neokortex, 1996
- [8] GROTH, P., GERBER, D.: Hands on SQL: the language, querying, reporting and the marketplace. 1.vyd. Prentice Hall Computer Books 1996, ISBN 0-234-86143-4
- [9] PRATT, P. J.: A guide to SQL. 3. vyd. North-Western 1995, ISBN 0-877-09520-5
- [10] DATE, C.J, WHITE, C.J.: PHP3 manual. <http://www.php.net>
- [11] WELSH, M., KAUFMAN, L.: Používáme Linux. 1. vyd. Computer Press 1997, ISBN 80-7226-001-4
- [12] FOTR, J.: Podnikatelské a investiční rozhodování. 1. Vyd. Praha 1995, Grada Publishing, ISBN 80-85623-20-X
- [13] BALLANTINE J., STRAY S.: Journal of Information technology 1998/13, Financial appraisal and the IS/IT investment decision making process

SEZNAM PŘÍLOH

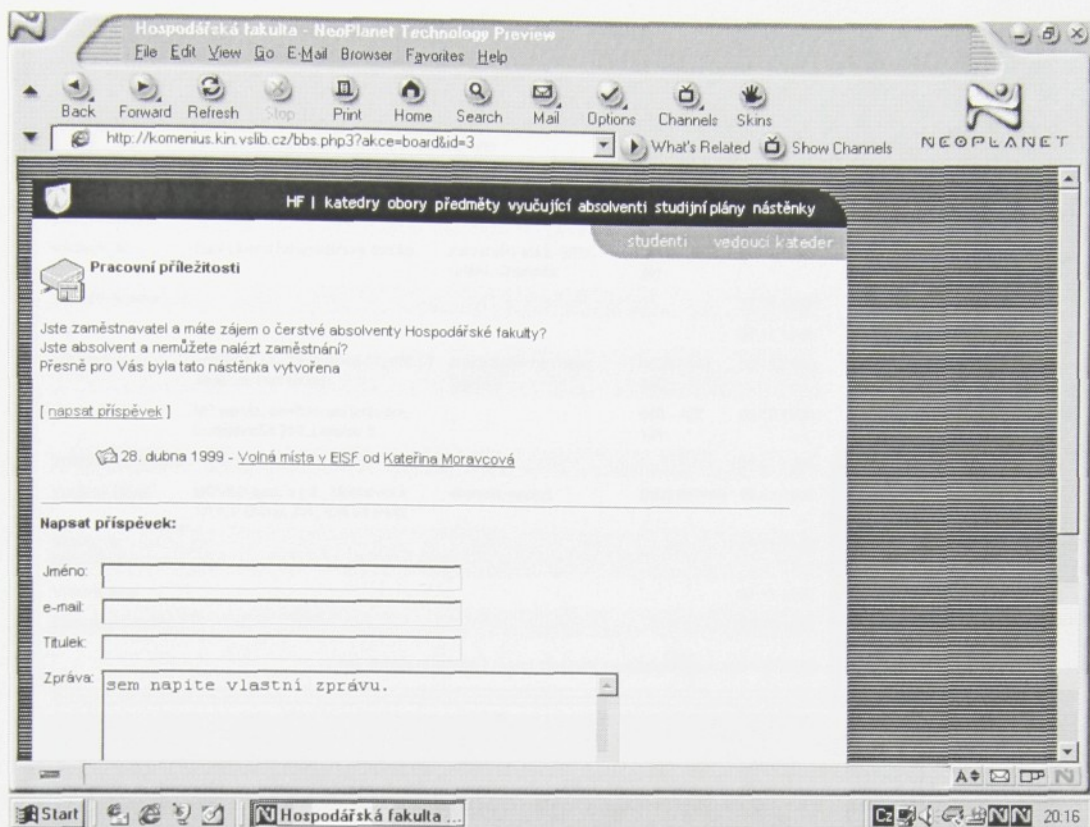
1. Ukázky uživatelského rozhraní (3 strany)



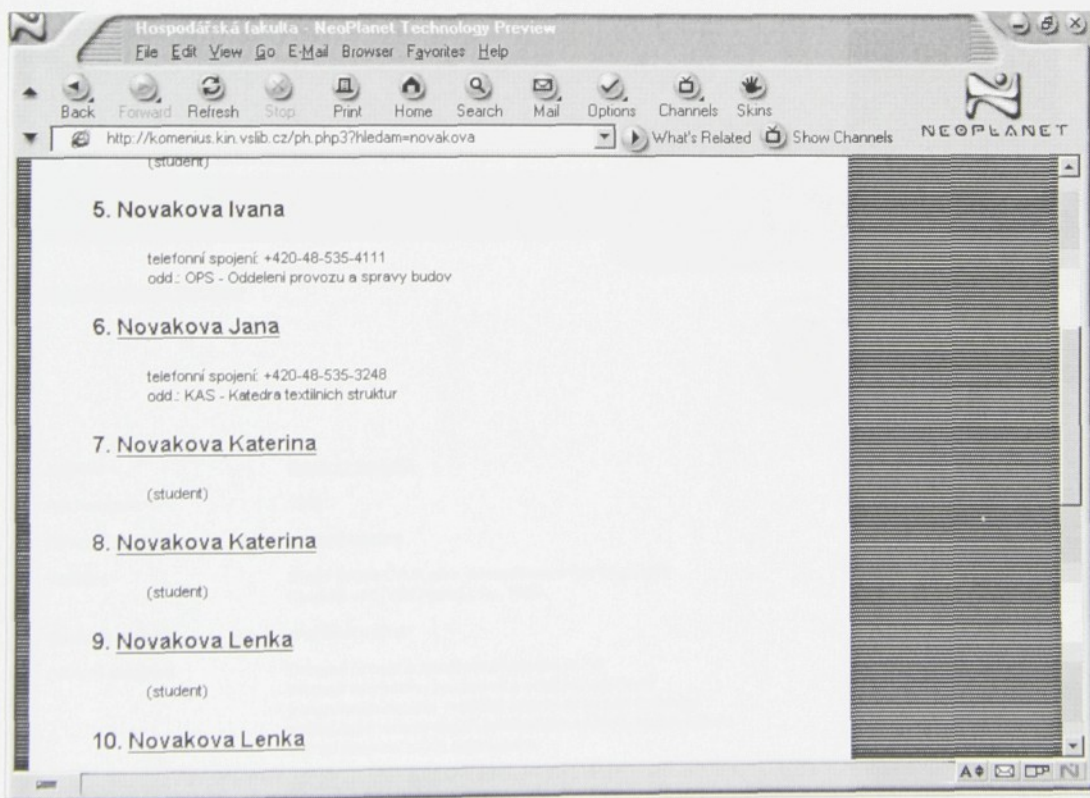
obrázek 1 - autorizace správce katedrových dat



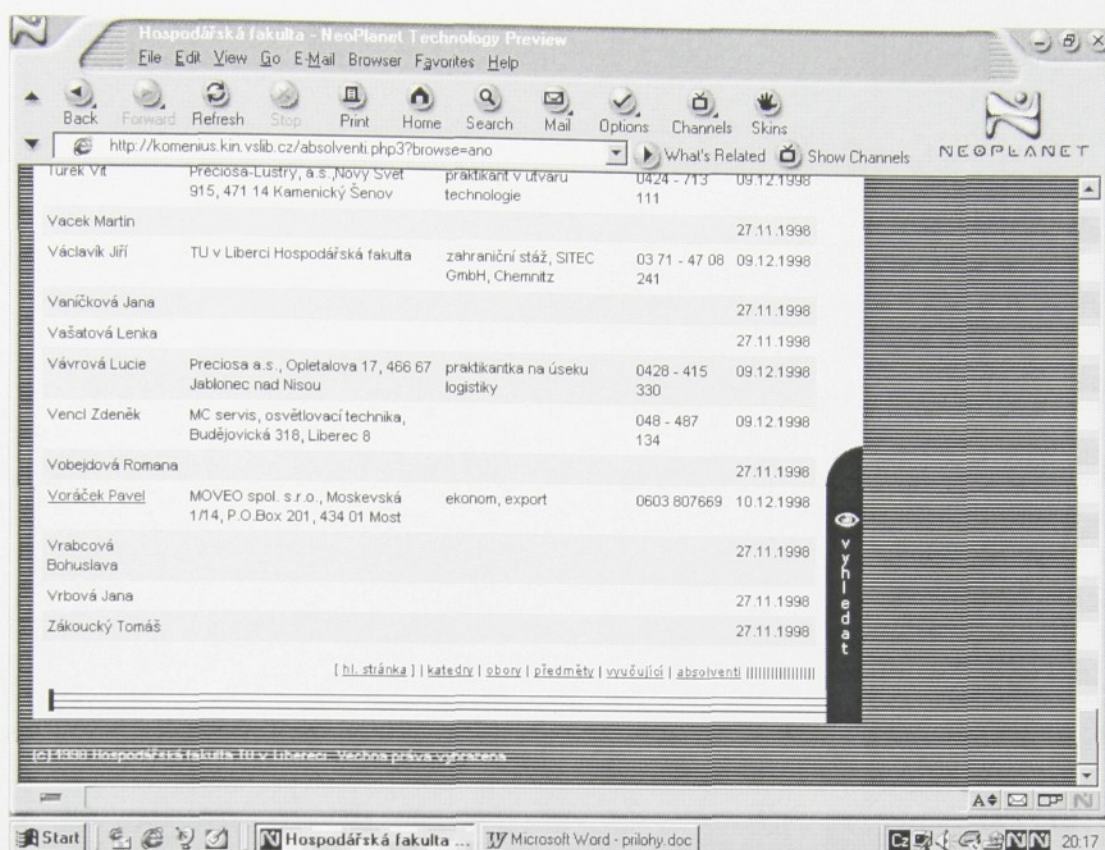
obrázek 2 - registrace volitelných předmětů



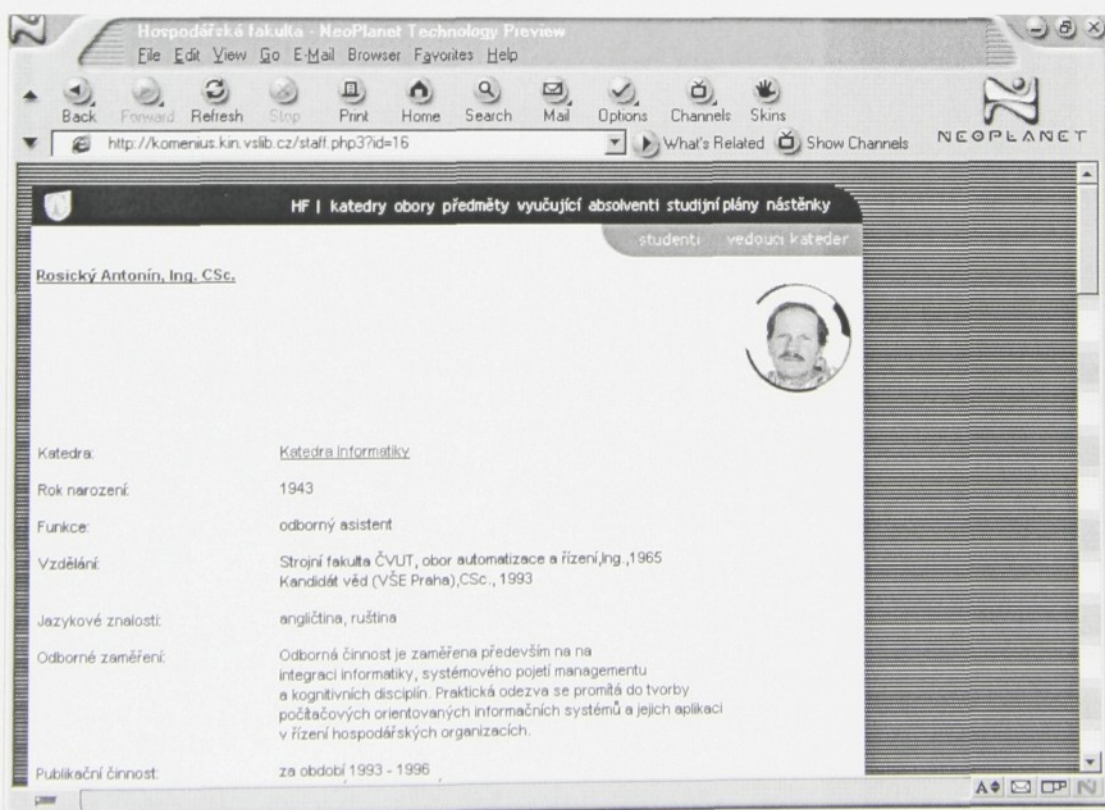
obrázek 3 - nástěnky HF



obrázek 4 - výsledek vyhledávání osob



obrázek 5 - výpis absolventů HF



obrázek 6 - informace o vyučujícím